



ESCUELA TÉCNICA SUPERIOR EN INGENIERÍA INFORMÁTICA  
GRADO EN INGENIERÍA INFORMÁTICA

Desarrollo de un simulador web para la representación de objetos en 3D.

Development of a web simulator that representates 3D objects.

Realizado por  
**Francisco Fernández Montiel**  
Tutorizado por  
**Rafael Marcos Luque Baena**  
Departamento  
**Lenguajes y Ciencias de la Computación**

UNIVERSIDAD DE MÁLAGA  
MÁLAGA, septiembre 2021

Fecha defensa: octubre 2021

## **Agradecimientos**

Mis agradecimientos van hacia mi tutor del TFG Rafa, que ha mostrado mucho empeño en que saliese como resultado un buen trabajo y a Jesús, Javi y Pablo que me han acompañado durante los cuatro años de carrera y nos hemos ayudado mucho en diversas situaciones en las que incluyo la motivación para realizar este proyecto.



## Resumen:

En este trabajo de fin de grado se presenta una alternativa a la manera tradicional de acceder a un catálogo online de una tienda de muebles y accesorios. Normalmente cada tienda pone a disposición de los clientes un archivo *PDF* en el que se puede visualizar los muebles a la venta con algunas de sus características. Debido a este modelo anticuado de dar a conocer los productos a los compradores surge la idea de una aplicación web que ofrece acceder a un catálogo el cual permite visualizar un modelo en 3D que nos ofrece la posibilidad tener una mejor representación del producto ampliando y rotando el mueble y modificarlo a nuestro gusto cambiando las texturas y materiales permitidas para dicho modelo.

El *Back-End* de la aplicación web estaría hecha con *Node* y *Express*, la base de datos *SQL* que se usa es *MariaDB*, el *Front-End* de la página estaría diseñada con *HTML* y *CSS* y para el configurador 3D se usa la herramienta para *WebGL* de *Three.js*. Como se puede observar, el lenguaje de programación a destacar es *JavaScript*, que hoy en día tiene una gran importancia en las labores de creación de aplicaciones web.

La aplicación también permitiría al administrador acceder a un área privada mediante un usuario y contraseña donde verá información personal y podrá observar aquellos encargos realizados por los clientes.

**Palabras claves:** *Node*, *Express*, *Three.js*, *MariaDB*, *JavaScript*, *API Rest*, Catálogo de muebles, *WebGL*, Configurador 3D, Aplicación Web.

---

**Abstract:**

This thesis presents an alternative to the traditional way of accessing an online catalogue of a furniture and accessories shop. Normally, each shop provides customers with a PDF file in which they can view the furniture on sale with some of its features. Due to this antiquated model of showing the products to the buyers, the idea of a web application arises that allows access to a catalogue which allows us to visualise a 3D model that offers us the possibility of having a better representation of the product by enlarging and rotating the furniture and modifying it to our liking by changing the textures and materials allowed for this model.

The Back-End of the web application is made with Node and Express, the SQL database used is MariaDB, the Front-End of the page is designed with HTML and CSS and for the 3D configurator I use the Three.js WebGL tool. As can be seen, the programming language to be highlighted is JavaScript, which nowadays is of great importance in the creation of web applications.

The application would also allow the administrator to access a private area with a username and password where he/she would see personal information and be able to view orders placed by customers.

**Keywords:** Node, Express, Three.js, MariaDB, JavaScript, API Rest, Furniture catalogue, WebGL, 3D configurator, Web App.



---

# Índice de contenidos

---

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	2
1.2. Objetivos . . . . .	3
1.3. Tecnologías y herramientas utilizadas . . . . .	4
1.3.1. Back-End . . . . .	4
Node.js . . . . .	4
Express . . . . .	4
Bcrypt . . . . .	4
Passport.js. . . . .	5
1.3.2. Bases de datos . . . . .	6
MariaDB . . . . .	6
HeidiSQL . . . . .	6
1.3.3. Front-End . . . . .	7
Three.js . . . . .	7
Bootstrap . . . . .	7
Docdroid . . . . .	7
Font-awesome . . . . .	8
JQuery . . . . .	8
1.3.4. IDE y editores de texto . . . . .	9
Visual Studio Code . . . . .	9
Notepad++ . . . . .	9
Overleaf . . . . .	9
1.3.5. Gestión de tareas y planificación . . . . .	10
Google Calendar . . . . .	10
1.3.6. Control de versiones . . . . .	10
Git . . . . .	10
1.3.7. Herramientas de almacenamiento . . . . .	11
GitHub . . . . .	11
1.3.8. Diagramas . . . . .	11
Lucidchart . . . . .	11
1.3.9. Lenguajes de programación . . . . .	12
JavaScript . . . . .	12
HTML . . . . .	12
CSS . . . . .	12
Python . . . . .	13
1.3.10. Conversores . . . . .	13
Convertio.co . . . . .	13

1.4. Estructura de la memoria . . . . .	14
<b>2. Estado del arte</b>	<b>15</b>
2.1. Lago . . . . .	15
2.1.1. Ventajas . . . . .	15
2.1.2. Desventajas . . . . .	15
2.2. Treku . . . . .	16
2.2.1. Ventajas . . . . .	16
2.2.2. Desventajas . . . . .	16
2.3. Cattelan Italia . . . . .	17
2.3.1. Ventajas . . . . .	17
2.3.2. Desventajas . . . . .	17
2.4. Lan Mobel . . . . .	18
2.4.1. Ventajas . . . . .	18
2.4.2. Desventajas . . . . .	18
<b>3. Desarrollo del proyecto</b>	<b>19</b>
3.1. Descripción general del proyecto . . . . .	19
3.2. Metodología de trabajo . . . . .	24
3.3. Análisis del proyecto . . . . .	26
3.3.1. Requisitos funcionales . . . . .	26
3.3.2. Requisitos no funcionales . . . . .	27
3.3.3. Casos de uso . . . . .	27
3.4. Fases de desarrollo . . . . .	33
3.4.1. Iteración 0: Análisis y diseño . . . . .	33
3.4.2. Iteración 1: Esqueleto de la aplicación . . . . .	33
3.4.3. Iteración 2: Configurador 3D . . . . .	33
3.4.4. Iteración 3: Catálogo y área privada . . . . .	34
3.4.5. Iteración 4: Corrección de errores e implementación de requisitos no funcionales . . . . .	34
3.4.6. Iteración 5: Finalización de la memoria . . . . .	34
3.5. Diseño del proyecto . . . . .	35
3.5.1. Base de datos . . . . .	35
3.5.2. Diseño del Backend . . . . .	37
3.5.3. Diseño del Frontend . . . . .	39
3.5.4. Diseño del configurador 3D . . . . .	41
3.6. Implementación . . . . .	42
3.6.1. El catálogo . . . . .	42
3.6.2. La base de datos . . . . .	44
3.6.3. Implementación del Front-End . . . . .	46
3.6.4. Implementación del Back-End . . . . .	50
3.6.5. Implementación del configurador 3D . . . . .	52
3.7. Pruebas . . . . .	54
3.7.1. Caso de prueba 1 . . . . .	54
3.7.2. Caso de prueba 2 . . . . .	54
3.7.3. Caso de prueba 3 . . . . .	55
3.7.4. Caso de prueba 4 . . . . .	55
<b>4. Conclusiones</b>	<b>57</b>



4.1. Conclusiones técnicas y aprendizaje personal . . . . .	57
4.2. Dificultades encontradas . . . . .	58
4.3. Futuras líneas de trabajo . . . . .	59
<b>Bibliografía</b>	<b>60</b>
<b>Apéndice A. Manual de Instalación</b>	<b>65</b>
A.1. Requisitos . . . . .	65
A.2. Paso para la instalación . . . . .	67
<b>Apéndice B. Manual de Usuario</b>	<b>69</b>
B.1. Manual cliente . . . . .	69
B.2. Manual administrador . . . . .	71



# CAPÍTULO 1

---

## Introducción

---

El hogar es, sin ninguna duda, uno de los bienes más preciados de las personas de nuestra sociedad dado que es el lugar donde pasamos la mayoría del tiempo y es el único sitio donde podemos tener una privacidad plena. Nuestra casa es parte de nosotros y se amolda a nuestros gustos y nuestra personalidad, de manera que cuando decidimos abrir la puerta a alguien, le estamos compartiendo algo personal que no compartiríamos con todo el mundo. Tampoco se puede hablar de hogar sin tener en cuenta los muebles que este lugar va a necesitar para darle utilidad, decoración y personalidad. Es aquí donde surge la necesidad que tiene la gente de adquirir un producto, adaptado al gusto de cada uno además de obtenerlo de manera sencilla y rápida. La manera tradicional de hacer esto es ir físicamente a buscar aquel mueble que tanto ansiamos, pero esto ya requeriría un desplazamiento y aun así no nos garantiza encontrarlo ya que, ¿qué persona no ha ido nunca a una tienda de muebles y no ha encontrado lo que andaba buscando? De este problema surge la idea de una aplicación web que nos permita visualizar un modelo 3D del mueble y poder personalizarlo a nuestro gusto y evitar desplazamientos innecesarios que, debido a la situación actual de la sociedad por el coronavirus, evitará posibles riesgos de contagio.

### 1.1. Motivación

Cuando decidí meterme en la carrera de ingeniería informática, nunca podría haber imaginado las posibilidades que me podría brindar aprender las distintas áreas de esta y la multitud de tecnologías y herramientas de las que podríamos hacer uso para desarrollar infinidad de proyectos diferentes. Y fue así que durante el paso de los 4 años en los cuales he estado estudiando, aprendiendo y descubriendo nuevas materias, hubo una que reunía varios componentes como las bases de datos, lenguajes de programación, uso de *frameworks*, diseño de páginas, etc. Con esto me refiero al área del desarrollo web y de todos sus componentes. Fue debido a esta diversidad lo que provocó que poco a poco me interesase y aprendiese *HTML*, *CSS* y *JavaScript*. Más tarde descubrí lo que eran los *Stacks* de desarrollo web y me dispuse a aprender *Node* y *Express*.

Todo surgiría meses antes de ver la propuesta de mi tutor del TFG, cuando vi una página web que incorporaba elementos 3D, me quedé fascinado y no dudé en investigar el modo en el que se podía incluir este tipo de elementos en archivos *HTML*. De alguna manera me pareció que era algo que iba a dar mucho de que hablar en el futuro, en el que la conexión a internet y las características de los nuevos ordenadores van a permitir que todo el mundo pueda cargar rápidamente en su navegador web estos modelos en 3D e interactuar con ellos y es que, hoy en día, nuestro mundo que está en constante evolución y se busca desarrollar todo aquello que pueda aportar utilidad y facilitar trabajos que con métodos tradicionales suponían más gasto energético para las personas.

Esto mencionado anteriormente es lo principal que he tenido en cuenta para el diseño de la aplicación web: utilizar una herramienta moderna y de gran utilidad, como es *Three.js* para incorporar elementos de *WebGl* a las vistas de la aplicación y ofrecer una visión más amplia y realista de como son, en este caso, los muebles que el cliente está interesado en adquirir. Algunas ventajas de este proyecto para el cliente es la posibilidad de visualizar desde casa el mueble que desea adquirir, personalizarlo para que el fabricante lo haga al gusto del cliente y el ahorro del viaje a una tienda física para ver muebles de manera realista.

Ahora si llegó el momento de elegir línea de trabajo y en cuanto vi la línea que ofertaba Rafael (mi tutor) no dudé en ningún momento, ya que un configurador 3D que use modelos de muebles me parece un escenario ideal y realista para empezar a utilizar *Three.js* como herramienta.

También me gusta mucho que este trabajo se aproxime mucho a lo que sería un proyecto real de una empresa, debido a que mi tutor me dio varios recursos sobre los que iba a trabajar y podría obtener esa experiencia de cara a conseguir un trabajo en el futuro en el que tendría que realizar mis tareas sobre proyectos empezados y tener la capacidad de adaptarme.

## 1.2. Objetivos

El objetivo de este proyecto es crear una aplicación web en la cual tengamos acceso a un catálogo de muebles y se pueda abrir un configurador del modelo en tres dimensiones del mueble para cambiar su materiales y texturas al gusto de cada uno y poder disponer de una mejor observación de dicho mueble aumentándolo y rotándolo. Es por ello, que los componentes de este trabajo son los siguientes:

- Front-End: componente donde se desarrollará la interfaz de la aplicación y donde se alojará el módulo del configurador de espacios en 3D.
- Back-End: donde se encontrará el código que genere el servidor, obtenga información de los módulos a representar y establezca la conexión con la base de datos.
- Módulo de simulación 3D usando *Three.js*: es parte del *Front-End* y va a ser el encargado de implementar todas las funcionalidades mencionadas anteriormente.

También se han añadido nuevos elementos al trabajo respecto a los iniciales, los cuales son:

- Catálogo de muebles: se ha creado un catálogo de muebles que nos permite acceder al configurador de cada modelo 3D y se ha integrado en la aplicación web.
- Área administradores: en la aplicación web se ha creado un apartado donde los administradores con un usuario y una contraseña pueden acceder y ver datos personales y consultar los encargos realizados por los clientes.

Con estos componentes nuestra aplicación será capaz de cumplir con todos los requisitos y de ser completamente funcional.

### 1.3. Tecnologías y herramientas utilizadas

#### 1.3.1. Back-End

##### Node.js

Es sin duda uno de los entornos de desarrollo web multiplataforma que más está dando de que hablar en los últimos años. *Node.js* es una opción ideal para poder indagar más a fondo y aprender sobre esta tecnología que usa como lenguaje de programación *JavaScript*, el cual también ha cobrado una gran importancia en las tareas relacionadas con el desarrollo web. Algunas características importantes de *Node* y más motivos por el que ha sido la elegida para este proyecto son:

- Es asíncrono y está controlados por eventos, por lo que nunca espera a que una *API* usada devuelva datos.
- Es muy rápida debido a la característica anterior y a que está favorecida por los principales navegadores al estar construido con *JavaScript*.
- Altamente escalable a pesar de que los procesos ocurran en un único hilo y licencia de código abierto lo que permite su libre uso de manera gratuita.



Figura 1.1: Node.[1]

##### Express

*Express* es un *framework* usado para *Node* y que facilita el desarrollo del *Back-End* de una aplicación web. Al igual que *Node*, fue lanzado como software gratuito y con licencia de código abierto.



Figura 1.2: Express.[2]

##### Bcrypt

Debido a que se va a emplear un método de inicio de sesión mediante y contraseña, iba a ser necesario reforzar la seguridad de la aplicación web y la primera herramienta se encontró para este propósito fue *Bcrypt*, que es una función de *hash* de contraseñas basada en la famosa función de *Blowfish*. Se eligió esta herramienta para el proyecto debido a su facilidad de integración con *Node* y la siguiente herramienta que se va a presentar.



Figura 1.3: Bcrypt.[3]

### Passport.js.

Como se acaba de mencionar, ya disponemos de una herramienta que se encarga de realizar la función *hash* de las contraseñas, pero adicionalmente es necesario otra herramienta que garantice la seguridad de la función de inicio de sesión y que pueda integrar *Bcrypt* e integrarse a si mismo en la aplicación. Es así que la herramienta elegida para el proyecto es *Passport.js* que es un *middleware* de autenticación para *Node.js* el cual tiene la capacidad de incorporarse de manera muy sencilla, flexible y escalable con cualquier aplicación basada en *Express*.



Figura 1.4: Passport.js.[4]

### 1.3.2. Bases de datos

#### MariaDB

En este proyecto se usa una base de datos *SQL* y se optó por *MariaDB*, que es un sistema de gestión de bases de datos derivado de *MySQL* y cuyo uso resulta muy sencillo y atractivo para realizar cualquier consulta o modificación.

Gracias a su motor Aria, que utiliza la caché en lugar de disco para almacenar lista de datos, permite realizar las consultas más complejas consiguiendo resultados a una gran velocidad. Al igual que las dos anteriores tecnologías, *MariaDB* también es un programa de código abierto y está distribuido bajo la licencia *GPL*.



Figura 1.5: MariaDB.[5]

#### HeidiSQL

Es un software de código libre y código abierto que con una interfaz muy amigable para el usuario y de uso intuitivo que permite establecer conexión con distintas bases de datos como *MySQL*, *PostgreSQL* o, en el caso de este proyecto, *MariaDB*.



Figura 1.6: HeidiSQL.[6]



### 1.3.3. Front-End

#### Three.js

Es una librería desarrollada en *JavaScript* que permite de manera fácil y eficiente añadir escenarios, elementos 3D y otros componentes a nuestra aplicación web dentro del navegador. Puede ser usada junto al elemento canvas de *HTML5*, *SVG* y *WebGL*. El código está abierto a todo el mundo y se encuentra en un repositorio de *GitHub*.

Es una librería que se está popularizando entre los desarrolladores y de la cual se espera un gran futuro gracias a la mejora de las conexiones y procesamiento de los nuevos dispositivos. Entre una infinidad de posibilidades, se puede destacar la creación de escenas, generación de luces ambientales, carga de modelos en 3D, uso de texturas y materiales, animaciones, etc.

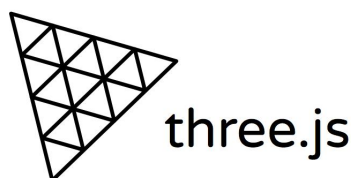


Figura 1.7: Three.js.[1]

#### Bootstrap

*Bootstrap* es una librería basada en *HTML*, *CSS* y *JavaScript* que nos permite de una manera muy sencilla y óptima modificar el diseño de nuestra página web. Es por ello que está enfocado al 100 % en el *Front-end* y es usado y conocido por la gran mayoría de desarrolladores de este campo de la informática.

Algunos de los muchos componentes que tiene *Bootstrap* son los siguiente: formularios, tablas, ajuste de columnas y líneas, tipografía, botones, cuadros, menús, etc. Es por el gran abanico de características y la simplicidad de su uso el motivo por el que se ha elegido para desarrollar el diseño web de este proyecto.



Figura 1.8: Bootstrap.[7]

#### Docdroid

La aplicación web requería que se mostrase el catálogo de los muebles en formato *PDF* y para ello se ha hecho uso de *Docdroid*, un servicio que nos permite subir y compartir

archivos y que también tiene un visor para *HTML* de dichos archivos y el cual se ha usado en la página principal de la aplicación.



Figura 1.9: Docdroid.[8]

### Font-awesome

*Font-awesome* es una librería para *HTML* y *CSS* que permite añadir logotipos e iconos a los textos de la página web y que se ha usado en el proyecto para convertir la interfaz más intuitiva para el usuario.



Figura 1.10: Font-awesome.[9]

### JQuery

*Jquery* es una herramienta muy extendida entre los desarrolladores y programadores en todo el mundo. Es una librería de *Javascript* que permite interactuar de diferentes maneras con documentos *HTML* modificando el *DOM Tree*, accionando y manejando eventos y otros tipos de interacciones.



Figura 1.11: JQuery.[10]

### 1.3.4. IDE y editores de texto

#### Visual Studio Code

Visual *Studio Code* es uno de los editores de código más extendidos en todo el mundo y para este proyecto era la mejor opción debido a todas las características que nos brinda. El diseño y la navegación entre archivos es muy cómoda, permite un acceso rápido a terminales, hay disponible una infinidad de *plug-ins* para descargar y mejorar la experiencia con el editor, integración con *Git*, etc.

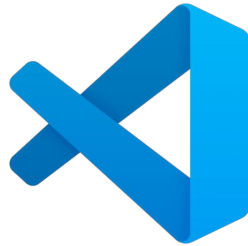


Figura 1.12: Visual Studio Code.[11]

#### Notepad++

Si es cierto que Visual *Studio Code* sería el editor de código principal, pero en muchas ocasiones se quiere abrir un archivo para realizar una consulta o hacer una edición rápida. Es en este tipo de ocasiones cuando uso *Notepad++*, ya que viene a ser parecido a un Bloc de notas, pero con ciertas características que lo asemejan a un editor de texto/código más elaborado y potente.

*Notepad++* también permite la instalación de extensiones, soporta diversos lenguajes y permite acciones de edición rápidas.

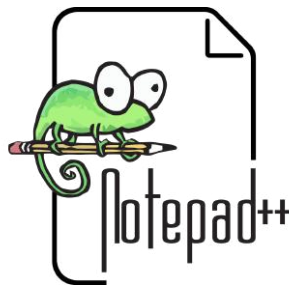


Figura 1.13: Notepad++. [12]

#### Overleaf

Como último editor de texto, voy a mencionar aquel en el cual se está realizando la memoria de este trabajo de fin de grado, *Overleaf*. *Overleaf* es un editor basado en la nube que nos permite crear proyectos de *LaTeX* y poder editar documentos de manera muy rápida e intuitiva. La característica que más me gusta de este editor es que permite el compilado de código a tiempo real y ver el archivo *.pdf* generado.



Figura 1.14: Overleaf.[13]

### 1.3.5. Gestión de tareas y planificación

#### Google Calendar

No hay mucho que decir de esta herramienta, como su nombre indica es la agenda y calendario *online* desarrollado por *Google* y brinda la posibilidad a sus usuarios de apuntar tareas y fechas importantes. Es una herramienta amada por muchos y se sitúa por encima de otros calendarios *online* por su simplicidad y la comodidad a la hora de acceder y tomar cualquier tipo de apunte.



Figura 1.15: Google Calendar.[14]

### 1.3.6. Control de versiones

#### Git

*Git* es un software de control de versiones que permite con comandos muy simples ir generando y almacenando nuevas versiones de nuestro código, de manera que si llega un punto en el que se comete un error en el desarrollo o se quiere hacer una consulta del código en una versión antigua, se puede recuperar muy fácilmente. A día de hoy, *Git* es el sistema de control de versiones más extendido en el mundo.



Figura 1.16: Git.[15]

### 1.3.7. Herramientas de almacenamiento

#### GitHub

Es una de las plataformas con más código libre alojado del mundo. *GitHub* es el repositorio de almacenamiento de *Git* y permite almacenar y guardar los proyectos de manera pública o privada. De esta plataforma hay que destacar la comunidad de desarrolladores que se ha formado y que diariamente comparten sus conocimientos y proyectos con el resto del mundo.



Figura 1.17: GitHub.[16]

### 1.3.8. Diagramas

#### Lucidchart

Para dar claridad y hacer más atractiva la memoria del proyecto, era necesario complementar la información con diagramas específicos para cada sección. Buscando herramientas gratuitas para hacer esto se encontró con *Lucidchart* y pareció una herramienta ideal para el propósito del trabajo debido a su simplicidad, pero a su vez debido a su alto grado de configuración y posibilidades que proporciona. Aunque no ha sido el caso de este trabajo, esta herramienta de diagramación permite trabajo colaborativo en línea y almacenar proyectos, por lo que resulta muy útil para proyectos grupales.



Figura 1.18: Lucidchart.[17]

### 1.3.9. Lenguajes de programación

#### JavaScript

*JavaScript* es el lenguaje de programación más extendido entre los desarrolladores web y de los más utilizados a día de hoy a nivel mundial. Uno de los principales motivos por el que este lenguaje es tan importante es porque su compilado es permitido por los principales navegadores web y permite dinamismo en las vistas de aplicaciones, es el lenguaje en el que se basan diversos *frameworks* de desarrollo y se define como un lenguaje orientado a objetos, de débil tipado, imperativo y basado en prototipos. En esta aplicación se usa para desarrollar la *app* con *Node* y *Express* y en las vistas para diferentes funcionalidades.

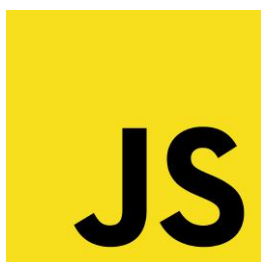


Figura 1.19: JavaScript. [18]

#### HTML

Es el lenguaje que se utiliza para diseñar las vistas de las aplicaciones web. Su rol es el de proporcionar un esqueleto a la página y su funcionamiento está basado en etiquetas de hipertexto y que se suele combinar con *CSS* para mejorar el diseño y con *JavaScript* por los motivos mencionados anteriormente. En esta aplicación se ha usado para generar la estructura principal de mis vistas.



Figura 1.20: HTML.

#### CSS

Es un lenguaje de diseño gráfico que nos permite añadir estilos y mejorar el diseño de las páginas *HTML*. En los últimos años este lenguaje se ha desarrollado mucho y ha sacado nuevas versiones que permiten generar animaciones y elementos gráficos dinámicos que están adquiriendo una gran popularidad. Es lenguaje junto a *CSS* se han usado exclusivamente en el *frontend* de la aplicación.



Figura 1.21: CSS.

## Python

Es el lenguaje más extendido en el mundo que destaca por la simpleza, eficiencia y legibilidad del código. Es un lenguaje multiparadigma ya que permite programación orientada a objetos, imperativa, funcional e imperativa. Debido a su eficiencia el rol que ha tenido este lenguaje en este proyecto ha sido para hacer diferentes pruebas con ficheros o componentes y también para la creación del *script* que añade los *links* al catálogo de los muebles.



Figura 1.22: Python. [19]

### 1.3.10. Conversores

#### Convertio.co

*Convertio.co* es una herramienta muy extendida utilizada para convertir ficheros de un formato específico a otro siempre que pueda existir una compatibilidad. En este proyecto se ha utilizado para convertir el catálogo *pdf* a *html* y posteriormente revertir ese cambio.



Figura 1.23: Convertio.co.

### 1.4. Estructura de la memoria

- **Capítulo 1: Introducción**

Durante este capítulo se han introducido los conceptos principales en los que se basará el proyecto, así como su motivación. También se han introducido las tecnologías utilizadas durante el desarrollo del proyecto desde su concepción, pasando por la implementación la página, hasta incluso la redacción de la propia estructura memoria.

- **Capítulo 2: Estado del arte**

En este capítulo se hace un análisis y un estudio de aplicaciones similares que se encuentran en el mercado. Del análisis realizado se exponen ventajas y desventajas de cada aplicación, para que en el desarrollo de este trabajo se pueda tener en cuenta las posibles debilidades y fortalezas para obtener un mejor producto.

- **Capítulo 3: Desarrollo del proyecto**

Inicialmente se hace una breve descripción del proyecto en el que se habla de todos los componentes que tendrá el mismo, e introduciremos la metodología ágil utilizada. También se hará un análisis del proyecto para establecer los requisitos no funcionales y casos de uso de la aplicación, para a continuación definir el diseño de la solución, tratando las distintas capas desde la base de datos, el configurador 3D y el diseño backend y frontend.

Después se llevará a cabo la implementación final del proyecto de forma iterativa para dar lugar a la página web completa, en esta parte se explicará en mayor profundidad cómo se han programado todos los componentes de la aplicación.

Finalmente, se expondrán algunas pruebas de las funcionalidades implementadas en el proyecto

- **Capítulo 4: Conclusiones**

En este último capítulo se analizarán las posibles líneas futuras de trabajo por donde llevar el proyecto resultante, así como los problemas técnicos encontrados, y las conclusiones obtenidas durante el desarrollo del TFG.



## CAPÍTULO 2

---

### Estado del arte

---

En esta sección se van a analizar diferentes configuradores 3D de tiendas de muebles que cumplen con características muy parecidas a las definidas para este proyecto y voy a definir puntos positivos y negativos para cada uno de tal modo que pueda tener un conocimiento más amplio sobre este sector y desarrollar una mejor aplicación:

#### 2.1. Lago

La empresa Lago tiene a su disposición su propio configurador en tres dimensiones en la cual se pueden cargar una gran diversidad de muebles en un fondo blanco y permite personalizarlo cambiando los materiales de dicho mueble y hacer *zoom* y rotarlo. De manera muy cómoda también permite cambiar de mueble sin dejar el configurador y exportarlo como *PDF* o modelo *OBJ*.

##### 2.1.1. Ventajas

- Diseño muy agradable, navegación intuitiva y simple, pero elegante.
- La calidad de los modelos y de las texturas son muy buenas.
- Permite exportar el modelo 3D.

##### 2.1.2. Desventajas

- Tarda en cargar los modelos y en cargar las texturas.
- No permite introducir varios muebles a la escena.



Figura 2.1: Configurador 3D de Lago.[20]

## 2.2. Treku

Treku es una tienda de muebles *online* que también ha querido dar el paso introduciendo a su aplicación web un configurador para añadir elementos. En este caso se usa un sistema de filtrado de muebles, indicando tipo, medidas y familia. Entre sus elementos destacan un botón para cambiar la vista de 2D a 3D y viceversa, botones para ampliar y rotar la escena y una tabla con diferentes texturas para aplicar al modelo.

### 2.2.1. Ventajas

- Carga rápida de los modelos.
- Se pueden incorporar varios muebles a la escena.

### 2.2.2. Desventajas

- Página poco atractiva y nada intuitiva.
- El diseño de los modelos son muy poco estéticos y no se asemejan a la realidad.
- Las texturas son poco realistas y aplicadas a las texturas pierden calidad.
- La tabla de texturas es muy extensa y no queda muy claro cuáles se pueden aplicar al modelo.

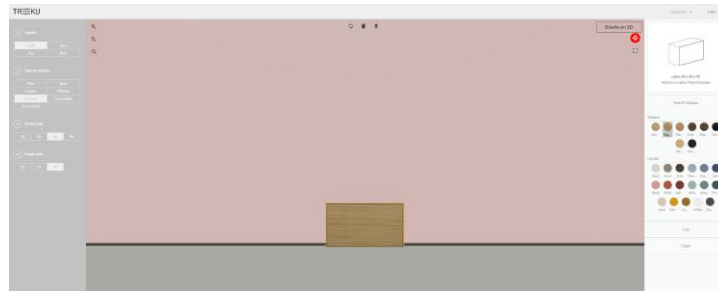


Figura 2.2: Configurador 3D de Treku.[21]

## 2.3. Cattelan Italia

Cattelan Italia destaca por tener en su catálogo un gran número de muebles y de gran calidad incorporando materiales como mármoles y otros tipos de piedras bien valoradas. La página es muy sencilla, tiene un sistema mediante el cual se puede filtrar por categorías y buscar de manera muy intuitiva el mueble deseado. Una vez se ha elegido un mueble se selecciona y se abre una ventana modal en la cual se puede cambiar las texturas del modelo.

### 2.3.1. Ventajas

- Página muy atractiva e intuitiva.
- Gran cantidad de muebles a elegir.
- Buen sistema de búsqueda y de filtrado de muebles.
- La ventana modal del configurador se abre rápidamente.

### 2.3.2. Desventajas

- Las texturas no quedan muy realistas una vez aplicadas.
- No permite cargar varios modelos.

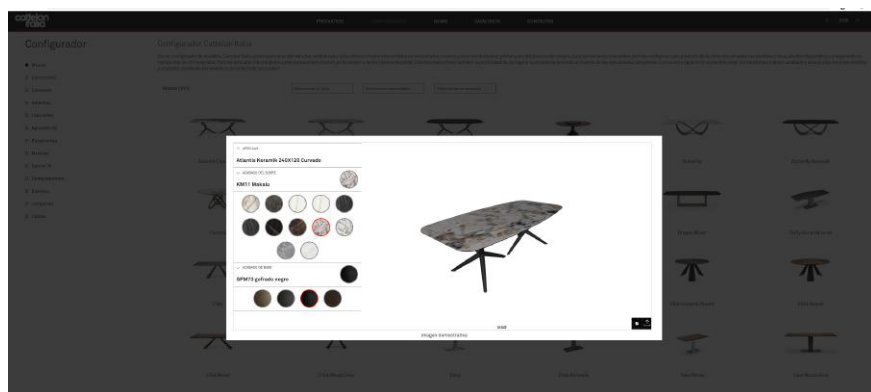


Figura 2.3: Configurador 3D de Cattelan Italia.[22]

## 2.4. Lan Mobil

Esta tienda de muebles posee un configurador que representa una habitación y permite cargar diferentes modelos en ella. Este configurador da un paso más allá y permite modificar las medidas de los muebles, así como la de las escenas y también permite dibujar los muros de la habitación virtual. Su interfaz se definiría más técnica que los otros configuradores mencionados y no parece estar enfocado a un cliente estándar.

### 2.4.1. Ventajas

- Gran cantidad de posibilidades de diseño de escena y muebles.
- Permite carga de varios modelos.
- Ofrece la posibilidad de ampliar y rotar la escena.
- Contiene escenas predefinidas para acelerar el proceso de edición.

### 2.4.2. Desventajas

- Enfocado a un perfil más técnico, por lo que la interfaz es poco intuitiva.
- Aplicar las texturas al modelo puede resultar algo complicado.
- Poca cantidad de muebles para personalizar.

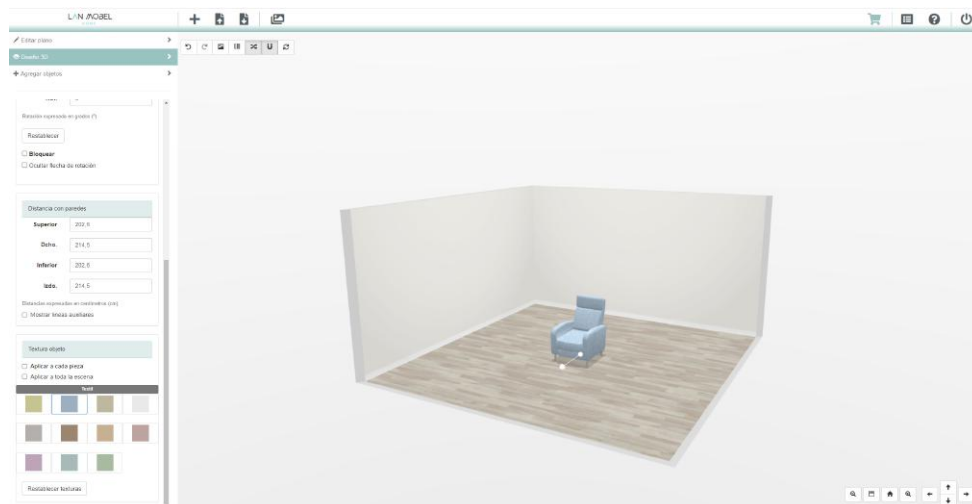


Figura 2.4: Configurador 3D de Lan Mobil.[23]

Tras analizar estos configuradores, se intentará adquirir para la aplicación de este proyecto todos aquellos componentes que se consideran ventajosos y se obtendrá un nuevo producto nuevo y mejor respecto a los ya existente. También hay que ser realista y los proyectos mencionados han sido desarrollados por personas experimentadas en un largo periodo de tiempo, así que se intentará aun teniendo poca experiencia implementar todas las características se definan y obtener una buena aplicación.

## CAPÍTULO 3

---

### Desarrollo del proyecto

---

#### 3.1. Descripción general del proyecto

Como ya se explica anteriormente, en un escenario real, este proyecto estaría destinado a una tienda de muebles con el objetivo de modernizar y proporcionar al cliente una nueva experiencia y brindarle la opción de visualizar un modelo en 3D del mueble deseado y permitir diversas acciones que ofrezcan una mejor visualización del mueble y un cambio del aspecto del mismo modificando materiales y texturas.



Figura 3.1: Portada App.

Por lo tanto, para acercar este trabajo a un escenario real se ha proporcionado al comienzo del proyecto una base de datos relacional bien estructurada de una tienda de muebles real. Esta base de datos contiene información de cada mueble, a que familia pertenecen, las texturas posibles para cada pieza y las relaciones de todas las tablas. Además, se ha podido obtener el catálogo de esa misma tienda de muebles en formato *PDF*, el cual se usa más adelante para añadirle enlaces al configurador 3D. Por último, también añadió un repositorio donde se encontraban almacenados todos los modelos 3D de los muebles

## CAPÍTULO 3. DESARROLLO DEL PROYECTO

en formato *.js* y las texturas que se pueden aplicar a los muebles.



Figura 3.2: Cuerpo App.

Para acceder a este configurador de modelos en tres dimensiones mencionado se pulsa en el enlace del mueble deseado que podemos encontrar en el catálogo, junto a la imagen de cada mueble. Este catálogo a su vez se encuentra integrado en una página web que simula el estilo que tendría una tienda de muebles *online*.

Para dar complejidad al trabajo, se ha añadido una sección en la página web en la cual los administradores, mediante un usuario y una contraseña, pueden acceder a un área privada en la cual se puede ver información personal, consultar los encargos realizados por los clientes y sirve de base para proyectos futuros como incluir un foro de empleados, añadir cómodamente nuevos muebles, hacer gestiones internas o, en general, un área donde poder ampliar la aplicación.

**➡ Iniciar Sesión**

**Usuario**  
si

**Contraseña**  
.....

**Recuérdame**  
☐

**Entrar**



[Volver a inicio.](#)

Figura 3.3: Pantalla de inicio de sesión.

**Bienvenido al area privada** [Cerrar Sesión](#) [Salir](#)

**Local**  
Usuario: ffernandez98  
Nombre completo: Francisco Fernández Montiel  
Edad: 23  
Puesto de trabajo: Administrador

**➡ Encargos de muebles**

<u>Nombre:</u> Cama	
<u>Descripción:</u> Cama con los siguientes materiales	
<u>Motivo:</u> Compra de cama	<u>Estado:</u> Materiales nuevos
	<a href="#">Eliminar encargo</a>
<u>Nombre:</u> Cabecera	
<u>Descripción:</u> Cabecera de madera	
<u>Motivo:</u> Alquiler de cabecera para apartamento	<u>Estado:</u> Reutilizado/nuevo
	<a href="#">Eliminar encargo</a>

[Crear nuevo administrador](#) [Crear](#)

Figura 3.4: Área privada de los administradores.

Una vez es elegido el mueble deseado, al pinchar en el código del mismo, se abrirá la pantalla del configurador 3D. En el centro de esta pantalla se observará el modelo en 3D del mueble que podremos rotar y ampliar con el ratón y, a la derecha de la pantalla, se podrá observar imágenes de las texturas asociadas a dicho modelo dividida en dos grupos: textiles y otros materiales. Para aplicar las texturas mostradas al modelo, se tendrá que pinchar en una de ellas, comprobando que sale un marco verde a la imagen y luego pinchar en la parte del modelo donde se quiera aplicar dicha textura.

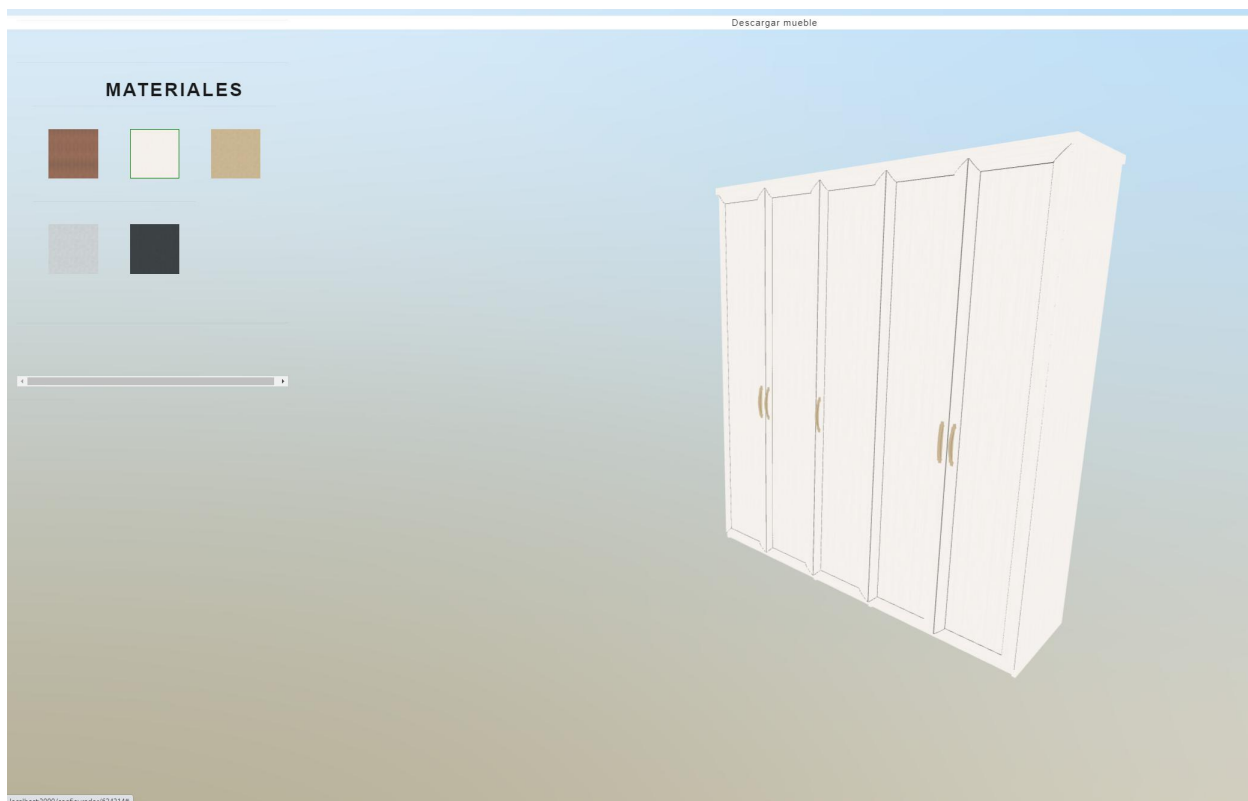


Figura 3.5: Configurador 3D.

Como se puede observar en la figura 2.5, vemos a la izquierda una tabla con las distintas texturas que se pueden aplicar a dicho modelo, en la parte central el modelo en 3D y en la parte superior un botón para descargar la imagen de dicho modelo.

Una vez el cliente ha elegido el mueble personalizado deseado, podrá descargarse una imagen de dicho mueble y encargarlo en el formulario que aparece en la página principal. En este formulario se deberá subir la imagen del mueble y concretar las especificaciones del encargo.

**➡ Añadir petición de encargo mueble**

Nombre

Descripción

Motivo de petición

Estado

Ningún archivo seleccionado

Figura 3.6: Formulario para encargar un mueble personalizado.



Para dar más realismo, no se permitirá aplicar las texturas en cualquier parte del mueble puesto que, por ejemplo, carece de sentido aplicar una textura de una madera en un cojín o un material que representa un espejo en las patas de un mueble. Así que ante este tipo de situaciones, si se intenta aplicar una textura a una zona no permitida, saltará una alerta de error y el cliente deberá llevar a cabo otra elección. La información de las texturas permitidas para cada sección del mueble viene recogida en las tablas y relaciones de la base de datos mencionada anteriormente.

Dos puntos que provocan que se piense así es el hecho de tratarse de un trabajo individual y que no habría terceros que dependiesen del desarrollo de esta parte o uno mismo no dependería de que otras personas cumpliesen sus plazos. Es por ello que este punto se puede resumir en mayor flexibilidad a la hora de realizar las diferentes tareas.

El siguiente punto y el más importante, es crear la aplicación desde una base y cumpliendo los requisitos básicos y según se fuesen implementando (estos requisitos iniciales) avanzar con funcionalidades más complejas y partes extra. De este modo se aseguraría cumplir con todos los plazos asignados, partiendo el desarrollo desde las funcionalidades más importantes hasta aquellas no tan importantes, y no se perdería tiempo en partes menos relevantes del trabajo. Es por ello que en las primeras iteraciones del desarrollo se va implementando las funcionalidades básicas y en las iteraciones finales se llevaría a cabo el desarrollo de las partes más complejas y funcionalidades extra para dar más valor al proyecto.

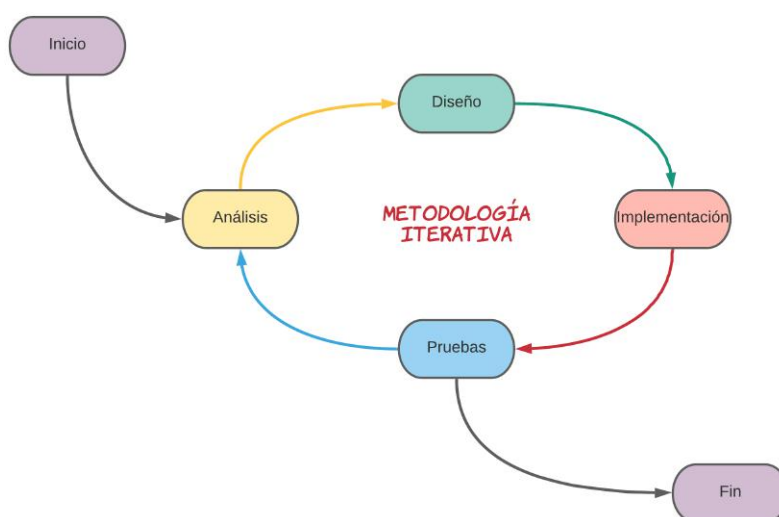


Figura 3.7: Metodología iterativa.

Es importante mencionar que al final de cada jornada de desarrollo del código se ha hecho uso continuo del sistema gestión de versiones *Git*[15], puesto que es posible que en algún momento pudiese surgir algún cambio en el desarrollo y con este sistema se podría volver a una versión anterior y continuar desde el punto de partida deseado. Y ya que se

ha mencionado *Git*, se tiene que mencionar también el que ha sido el repositorio del código durante todo el desarrollo, *GitHub*[16]. Aunque el código siempre ha estado almacenado en el ordenador personal, no se podía correr el riesgo de perder todo lo avanzado y es el motivo que hizo plantearse ir guardando en este repositorio *online* todo el código de la aplicación.

Por último, una herramienta a la que se ha dado uso continuo para organizarse y poder visualizar plazos y tareas pendientes ha sido *Google Calendar*[14], un calendario clásico *online* que ha permitido de manera muy cómoda ir apuntando todo lo que ha ido haciendo falta.

### 3.3. Análisis del proyecto

Antes de iniciar el desarrollo del proyecto, era necesario llevar a cabo un análisis de los requisitos y es por este motivo fue necesario concertar varias citas con el tutor del TFG para que dejase claro todos los puntos del proyecto y hacerse una idea clara de todos los requisitos que la aplicación debía cumplir para su correcto funcionamiento. Para este listado de requisitos, se va a hacer una división entre los funcionales, que definen las funciones del sistema, y los no funcionales, que son características y restricciones propias de la *app*.

Una vez definidos estos requisitos, se va a proceder a hacer un listado de los diferentes casos de uso de la aplicación, viendo así todas las aplicaciones que tiene el proyecto de manera más clara.

#### 3.3.1. Requisitos funcionales

Estos son los requisitos funcionales de la aplicación web acordados con el tutor:

- RF1: Existen varias piezas/muebles.
- RF2: Existen varias texturas.
- RF3: Cada mueble tiene asignadas unas texturas.
- RF4: Cada parte del mueble permite unas texturas determinadas.
- RF5: En el configurador 3D se puede cambiar las texturas al mueble.
- RF6: En el configurador 3D se puede ampliar y rotar el mueble.
- RF7: En el configurador 3D se puede descargar una imagen del mueble personalizado.
- RF8: Desde la aplicación web se puede acceder al catálogo de los muebles.
- RF9: Desde la aplicación web se puede rellenar un formulario para encargar un mueble.
- RF10: En cada mueble del catálogo dispondremos de uno o varios *links* para acceder al configurador de dicho mueble.
- RF11: En la aplicación web, habrá un apartado para acceder al área privada de los administradores.
- RF12: Al área privada de los administradores se accederá con un usuario y una contraseña.
- RF13: En el área privada se podrá ver información del administrador y ver los encargos realizados por los clientes.
- RF14: En el área privada el administrador podrá crear nuevos perfiles administradores.

### 3.3.2. Requisitos no funcionales

Los requisitos no funcionales que debe cumplir la aplicación son los siguientes:

- RNF1: Carga rápida tanto de la página principal de la web como del configurador 3D.
- RNF2: Seguridad de navegación y de uso de funcionalidades de la app.
- RNF3: Seguridad en la identificación del administrador y privacidad de datos.
- RNF4: Seguridad en la contraseña (se almacenará encriptada en la base de datos).
- RNF5: Se incluirá un manual de usuario para hacer uso de la aplicación.
- RNF6: Se dispondrá de un manual de instalación para instalar la app de manera local.
- RNF7: Facilidad de uso e instalación.

### 3.3.3. Casos de uso

A continuación se definen los casos de uso de la aplicación web:

Caso de Uso 1	Acceso catálogo.
Resumen	El usuario/administrador accede a la aplicación y en ella observa el catálogo.
Actor	Usuario/administrador.
Precondición	El usuario debe disponer del <i>link</i> de la página web
Escenario	<ul style="list-style-type: none"> <li>■ El usuario/cliente accede mediante un enlace a la página web.</li> <li>■ El usuario avanza hacia abajo por la página hasta llegar al catálogo de muebles.</li> <li>■ El usuario puede ampliar, recorrer o descargar el <i>PDF</i> del catálogo.</li> </ul>

Tabla 3.1: Caso de Uso 1.

Caso de Uso 2	Acceso configurador 3D.
Resumen	El usuario/administrador accede al configurador 3D de un mueble.
Actor	Usuario/administrador.
Precondición	El usuario debe disponer del <i>link</i> de la página web.
Escenario	<ul style="list-style-type: none"> <li>■ El usuario/cliente accede mediante un enlace a la página web.</li> <li>■ El usuario avanza hacia abajo por la página hasta llegar al catálogo de muebles.</li> <li>■ El usuario elige un mueble y hace <i>click</i> en el <i>link</i> que lleva al configurador 3D de dicho mueble.</li> </ul>

Tabla 3.2: Caso de Uso 2.

Caso de Uso 3	Cambiar textura de mueble.
Resumen	El usuario/administrador accede al configurador 3D de un mueble y le cambia una textura.
Actor	Usuario/administrador.
Precondición	El usuario debe disponer del <i>link</i> de la página web.
Escenario	<ul style="list-style-type: none"><li>■ El usuario/cliente accede mediante un enlace a la página web.</li><li>■ El usuario avanza hacia abajo por la página hasta llegar al catálogo de muebles.</li><li>■ El usuario elige un mueble y hace <i>click</i> en el <i>link</i> que lleva al configurador 3D de dicho mueble.</li><li>■ El usuario pincha en la textura deseada hasta que quede marcada y, a continuación, pincha en la parte del mueble donde quiere aplicarla.</li></ul>

Tabla 3.3: Caso de Uso 3.

Caso de Uso 4	Rotar y ampliar mueble.
Resumen	El usuario/administrador accede al configurador 3D de un mueble y lo rota y amplía/aleja a su gusto.
Actor	Usuario/Administrador.
Precondición	El usuario debe disponer del <i>link</i> de la página web.
Escenario	<ul style="list-style-type: none"><li>■ El usuario/cliente accede mediante un enlace a la página web.</li><li>■ El usuario avanza hacia abajo por la página hasta llegar al catálogo de muebles.</li><li>■ El usuario elige un mueble y hace <i>click</i> en el <i>link</i> que lleva al configurador 3D de dicho mueble.</li><li>■ El usuario pincha en el fondo del configurador web y lo mueve para rotar el modelo del mueble. Con la rueda del ratón puede ampliar y alejar el mueble.</li></ul>

Tabla 3.4: Caso de Uso 4.

Caso de Uso 5	Descargar imagen del mueble personalizado
Resumen	El usuario/administrador accede al configurador 3D y descarga una imagen del mueble personalizado
Actor	Usuario/Administrador.
Precondición	El usuario debe disponer del <i>link</i> de la página web y haber personalizado el mueble.
Escenario	<ul style="list-style-type: none"><li>■ El usuario/cliente accede mediante un enlace a la página web.</li><li>■ El usuario avanza hacia abajo por la página hasta llegar al catálogo de muebles.</li><li>■ El usuario elige un mueble y hace <i>click</i> en el <i>link</i> que lleva al configurador 3D de dicho mueble.</li><li>■ El usuario personaliza el mueble y le da al botón de descargar para obtener la imagen del mueble.</li></ul>

Tabla 3.5: Caso de Uso 5.

Caso de Uso 6	Acceder área privada.
Resumen	El administrador puede <i>logearse</i> y acceder a un área privada de administradores donde puede ver información personal.
Actor	Administrador.
Precondición	El administrador debe tener el enlace a la página web y disponer de un usuario y contraseña.
Escenario	<ul style="list-style-type: none"><li>■ El administrador accede mediante un enlace a la página web.</li><li>■ El administrador avanza hacia abajo por la página hasta llegar al botón para acceder a la pantalla de login.</li><li>■ El administrador introduce su usuario y contraseña.</li><li>■ Si es correcta accederá al área privada donde verá su información personal.</li></ul>

Tabla 3.6: Caso de Uso 6.

Caso de Uso 7	Añadir nueva cuenta de administrador
Resumen	El administrador puede <i>logearse</i> y acceder a un área privada de administradores donde puede crear un nuevo usuario administrador
Actor	Administrador.
Precondición	El administrador debe tener el enlace a la página web y disponer de un usuario y contraseña.
Escenario	<ul style="list-style-type: none"><li>■ El administrador accede mediante un enlace a la página web.</li><li>■ El administrador avanza hacia abajo por la página hasta llegar al botón para acceder a la pantalla de login.</li><li>■ El administrador introduce su usuario y contraseña.</li><li>■ Si es correcta accederá al área privada.</li><li>■ Creará nuevo usuario con rol de administrador</li></ul>

Tabla 3.7: Caso de Uso 7.

Caso de Uso 8	Consultar encargo de los clientes
Resumen	El administrador puede <i>logearse</i> y acceder a un área privada de administradores donde puede consultar los encargos realizados por los cliente
Actor	Administrador.
Precondición	El administrador debe tener el enlace a la página web y disponer de un usuario y contraseña.
Escenario	<ul style="list-style-type: none"><li>■ El administrador accede mediante un enlace a la página web.</li><li>■ El administrador avanza hacia abajo por la página hasta llegar al botón para acceder a la pantalla de login.</li><li>■ El administrador introduce su usuario y contraseña.</li><li>■ Si es correcta accederá al área privada.</li><li>■ Consultará una lista con los distintos encargos de los clientes, la imagen adjunta y sus especificaciones</li></ul>

Tabla 3.8: Caso de Uso 8.



A continuación se muestran los diagramas de casos de uso para cada uno de los actores, que en el caso de esta aplicación son el usuario y el administrador:

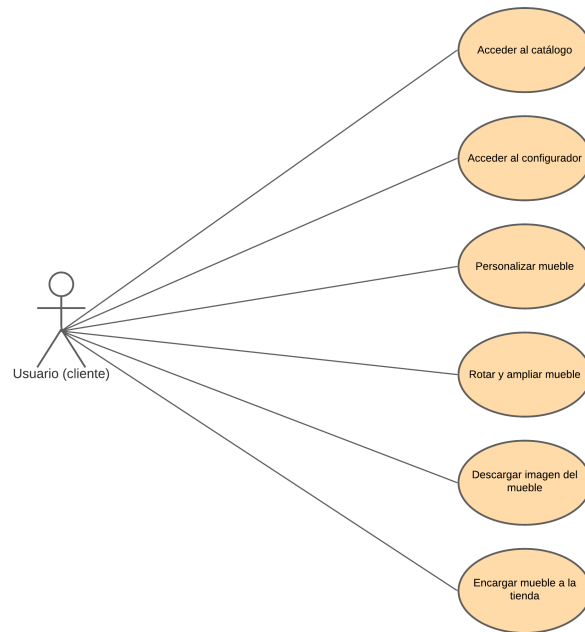


Figura 3.8: Diagrama de casos de uso del usuario.



Figura 3.9: Diagrama de casos de uso del administrador.

Finalmente, para conocer un poco como son aquellas acciones principales de los actores, se procede a representar en dos diagramas de secuencia que representan el proceso de encargar un mueble por parte del cliente y la acción de entrar en el área privada y visualizar encargos, registrar un nuevo administrador y eliminar encargos que ya se ha llevado a cabo:

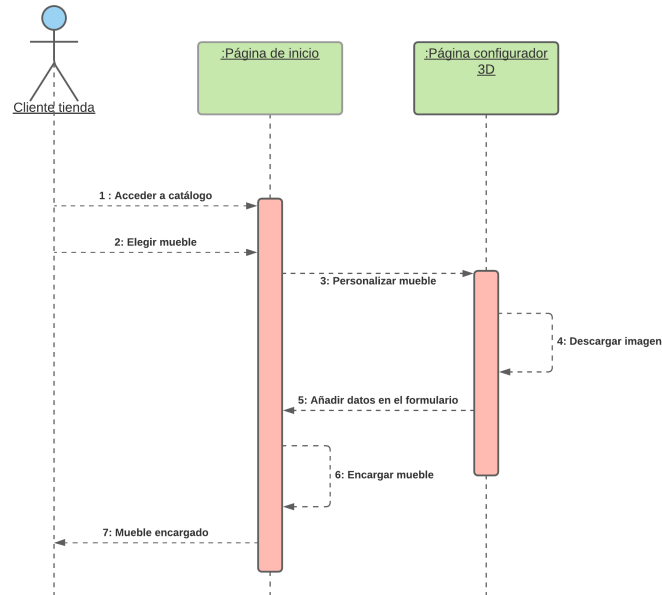


Figura 3.10: Diagrama de secuencia del usuario (cliente).

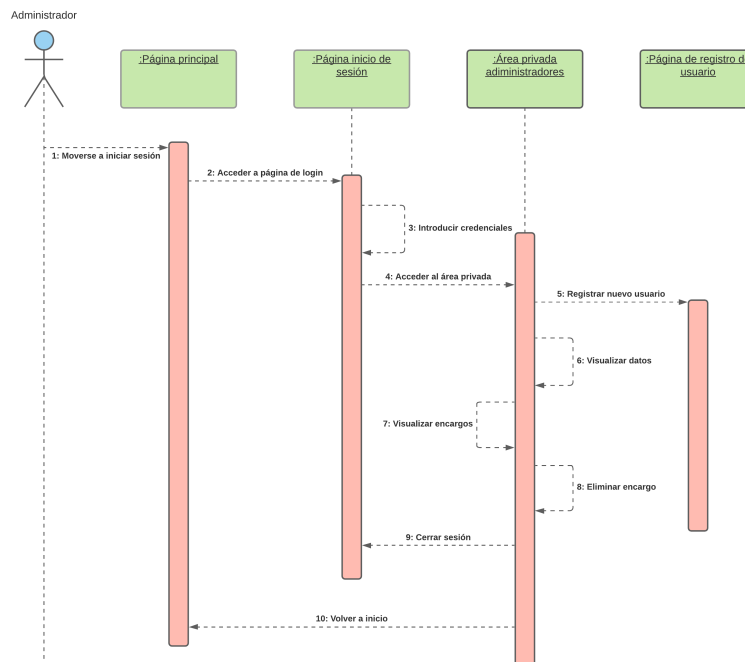


Figura 3.11: Diagrama de secuencia del administrador.

## 3.4. Fases de desarrollo

En esta sección se explica de manera muy superficial que se ha hecho en cada iteración y que requisitos se han cumplido.

### 3.4.1. Iteración 0: Análisis y diseño

Durante esta etapa del proyecto se contacta con el tutor y se realiza un análisis en profundidad para obtener los casos de uso, los requisitos funcionales y no funcionales con los que contará la aplicación. Acto seguido se realiza el diseño conceptual de la aplicación dónde se indica cómo se van a implementar estos requisitos, el aspecto y las funcionalidades de la aplicación.

Requisitos cumplimentados:

- RF1: Existen varias piezas/muebles.
- RF2: Existen varias texturas.
- RF3: Cada mueble tiene asignadas unas texturas.
- RF4: Cada parte del mueble permite unas texturas determinadas.

### 3.4.2. Iteración 1: Esqueleto de la aplicación

Se inicia la implementación del proyecto y se sincroniza con el repositorio de control de versiones creado en *GitHub*. En esta etapa utilizando *Node* se crea el esqueleto de la página web y se generan todos los ficheros y componentes necesarios.

Requisitos cumplimentados:

- RF8: Desde la aplicación web se puede acceder al catálogo de los muebles.
- RF9: Desde la aplicación web se puede rellenar un formulario para encargar un mueble.

### 3.4.3. Iteración 2: Configurador 3D

Durante esta etapa se desarrolla el configurador 3D en su totalidad con todas las funcionalidades propuestas. También se realizan modificaciones en la página web para proporcionar compatibilidad al configurador con el resto de elementos implementados tales como la base de datos.

Requisitos cumplimentados:

- RF5: En el configurador 3D se puede cambiar las texturas al mueble.
- RF6: En el configurador 3D se puede ampliar y rotar el mueble.
- RF7: En el configurador 3D se puede descargar una imagen del mueble personalizado.

### 3.4.4. Iteración 3: Catálogo y área privada

En esta etapa de desarrollo se genera el catálogo en *pdf* con los *links* que iniciarán el configurador con el modelo 3D del mueble seleccionado. Además se crea el área privada a la que tendrán acceso los administradores de la web con las funcionalidades definidas en el análisis y diseño del proyecto.

Requisitos cumplimentados:

- RF10: En cada mueble del catálogo dispondremos de uno o varios *links* para acceder al configurador de dicho mueble.
- RF11: En la aplicación web, habrá un apartado para acceder al área privada de los administradores.
- RF12: Al área privada de los administradores se accederá con un usuario y una contraseña.
- RF13: En el área privada se podrá ver información del administrador y ver los encargos realizados por los clientes.
- RF14: En el área privada el administrador podrá crear nuevos perfiles administradores.

### 3.4.5. Iteración 4: Corrección de errores e implementación de requisitos no funcionales

En esta etapa final de la implementación, se arreglan los problemas que hayan surgido durante el resto del desarrollo, se retocan los aspectos de diseño que hayan quedado obsoletos y se terminan de añadir los requisitos no funcionales que otorguen calidad a la página.

- RNF1: Carga rápida tanto de la página principal de la web como del configurador 3D.
- RNF2: Seguridad de navegación y de uso de funcionalidades de la app.
- RNF3: Seguridad en la identificación del administrador y privacidad de datos.
- RNF4: Seguridad en la contraseña (se almacenará encriptada en la base de datos).
- RNF5: Se incluirá un manual de usuario para hacer uso de la aplicación.
- RNF6: Se dispondrá de un manual de instalación para instalar la app de manera local.
- RNF7: Facilidad de uso e instalación.

### 3.4.6. Iteración 5: Finalización de la memoria

Finalmente una vez acabada el desarrollo del proyecto, se procede a dar los últimos retoques a la memoria.

### 3.5. Diseño del proyecto

Una vez realizado el análisis de los requisitos, tenía que plantearse como iba a ser el diseño de los diferentes componentes de la aplicación para que fuese completamente funcional, cumpliera todos los requisitos y que se caracterizase por la simpleza y experiencia de usuario. Antes de iniciar el diseño, se diseñó la arquitectura de la aplicación y se hizo el siguiente diagrama:

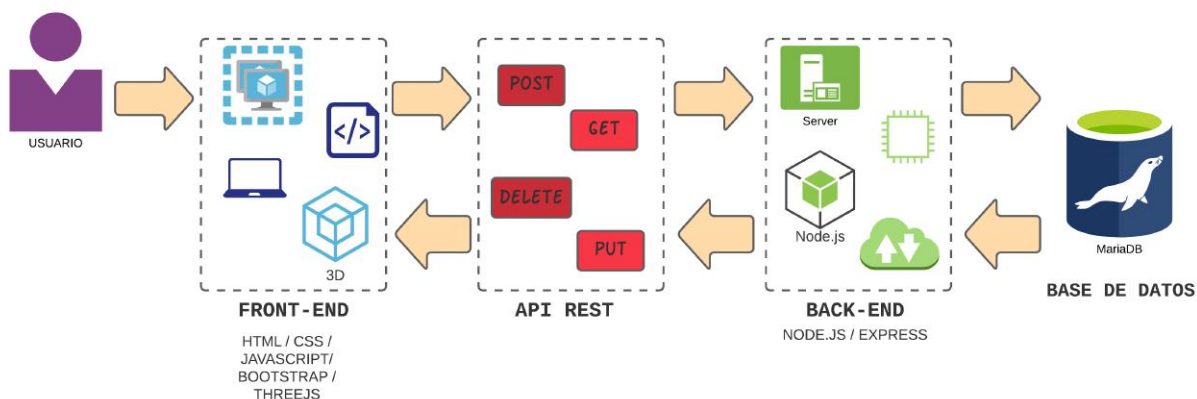


Figura 3.12: Diagrama de la arquitectura de la aplicación.

#### 3.5.1. Base de datos

Fue aportada al principio del proyecto, aunque se ha añadido algunos cambios como una tabla para almacenar la información de los administradores u otra para almacenar los encargos de muebles que los clientes realicen. Como se ha dicho, la mayor parte del contenido de la base de datos fue aportada antes de comenzar el proyecto, pero se considera que es importante echarle un ojo al diseño para poder construir el *Back-End* en torno a ella y cumplir todos los requisitos de la aplicación.

A continuación se muestra un diagrama entidad-relación de la base de datos del proyecto en la que las tablas “Administradores” y “PeticionMuebles” fueron añadidas adicionalmente a lo aportado al comienzo del proyecto y el resto de tablas aportadas previamente al comienzo del trabajo:

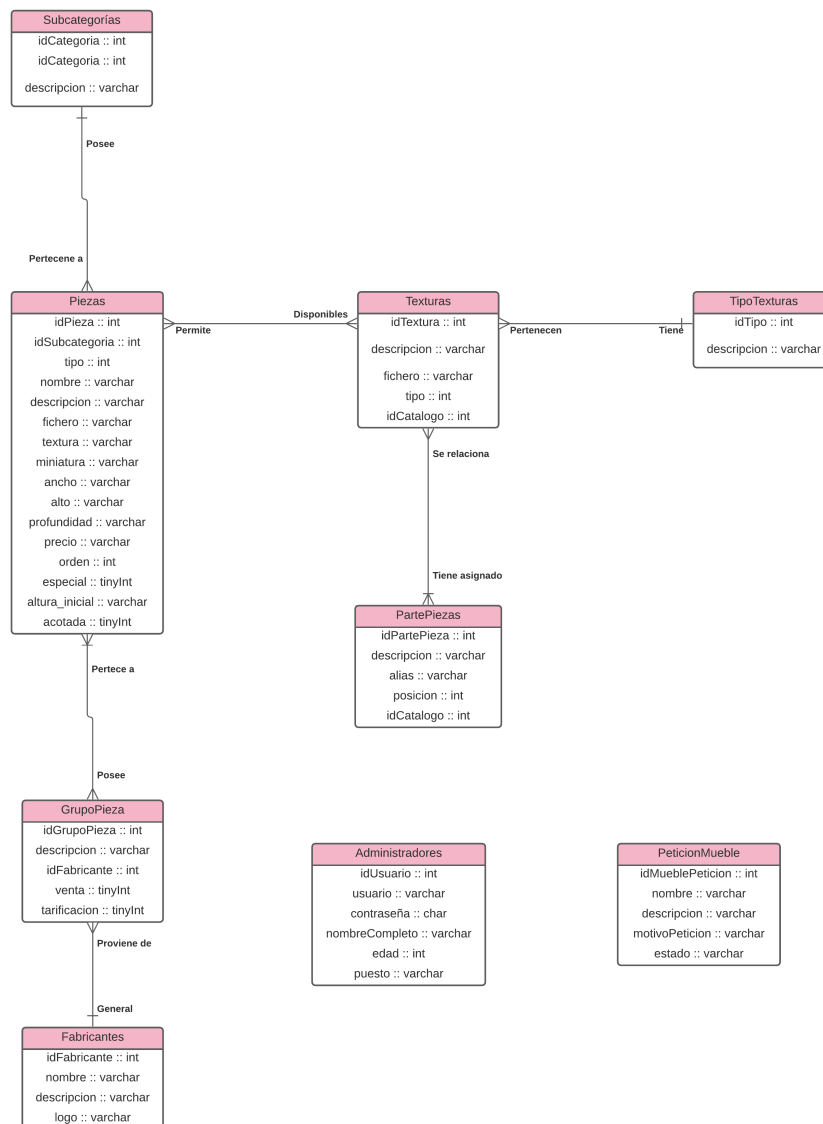


Figura 3.13: Diagrama Entidad-Relación de la base de datos.

### 3.5.2. Diseño del Backend

Como se menciona en la introducción de esta memoria, para la construcción del *Back-End* se decidió usar *Node* con *Express*[24], ya que era una tecnología que quería aprender a usar ya que en los últimos años está siendo muy demandada y me daba la impresión que me iba a resultar muy cómoda. Entonces, para la creación del servidor, se tenía que crear una *API Rest* que ofreciese todas las funcionalidades necesarias y que se integrase bien con la base de datos. La parte con la que se tenía que tener más cuidado es cuando se accede al configurador 3D, ya que son muchos datos muy específicos los que hay que enviar y se puede cometer algún error en cualquier momento.

Así pues tenía que pensar como se iba a hacer toda la estructura del *back-end* de la aplicación y se empezaron a crear todas las carpetas y archivos necesarios para el funcionamiento correcto del proyecto. Para ello se creó un diagrama de componentes de la parte del servidor y se empezó a trabajar sobre ella:

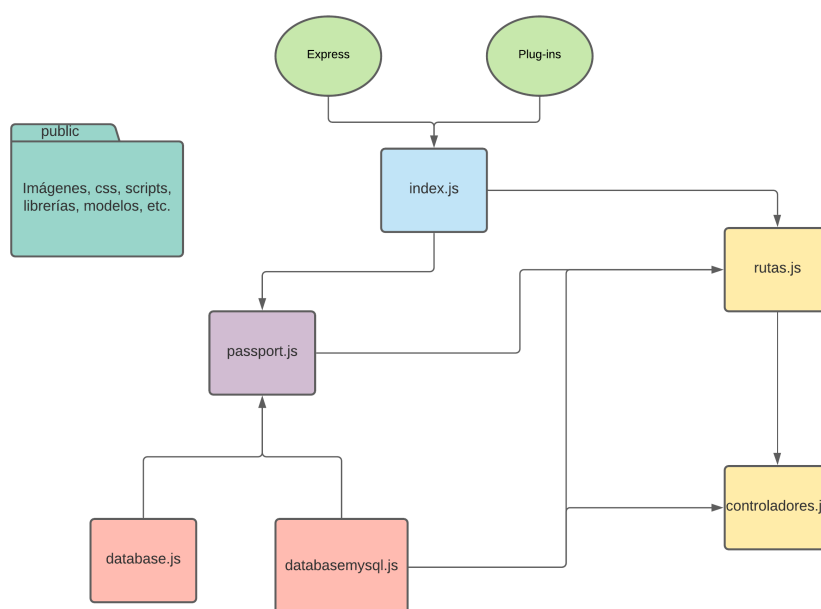


Figura 3.14: Diagrama de componentes del Back-End.

Primero se crearía el archivo principal de la aplicación en la cual albergaría todos los elementos relacionados con la configuración, que incluye la importación de los *plug-ins* utilizados, los *middlewares*, las definiciones del servidor y el acceso a este, la localización de los archivos de rutas, la definición del repositorio estático, etc. En resumen, todo lo necesario para echar a andar la aplicación.

Después necesitaría un archivo de configuración para conectar mi aplicación web con la base de datos. La base de datos utilizada deriva de *MySQL*, por lo que se debería encontrar el conector que permitiese dicha integración y de este modo ya se podría hacer uso de todas las tablas y datos del *database*.

Para permitir una correcta navegación y estructurar bien el proyecto, se crearía una carpeta de rutas con un fichero que contuviera todos los componentes y métodos para convertir el servidor vacío en una *API Rest*. Es por este motivo que los métodos que se usaría son *GET*, *POST*, *DELETE* y *PUT*. Estas rutas y sus respectivos controladores

(encontrados en el archivo de controladores del proyecto), permitirían la navegación entre páginas de las vistas, el envío de datos al *front-end*, ofrecer seguridad a la aplicación encriptando la contraseña del *login*, obtener datos de la base de datos y aplicar diferentes características y restricciones al proyecto.

Como se ha mencionado anteriormente, es necesario un repositorio situado en el lado del servidor para almacenar todo aquello necesario para satisfacer todos los requisitos y servir de repositorio que apoye al *front-end* con todos aquellos archivos, imágenes, librerías, modelos 3D, *scripts* y demás contenido que sea fundamental para el desarrollo.

Por último, debido a la necesidad de algunas librerías o componentes, resultaba fundamental la creación de una carpeta auxiliar de configuración donde ir metiendo archivos necesarios para las rutas y el controlador como un archivo con las características de la base de datos o un archivo necesario para el encriptado para la contraseña en la parte del login.

A rasgos generales esta es la idea de diseño del *back-end* para este trabajo de fin de grado, la cual se pretende que como característica principal sea escalable y esté bien estructurada. Esto se decidió que fuese así, ya que se asemeja a un proyecto realista en una empresa y quería ganar esa cantidad de experiencia. También se pensó que es un proyecto que puede dar mucho de sí y que aunque estuviese centrado en el configurador 3D, más adelante la aplicación podría crecer (y como proyecto personal lo hará) y tendrá que desarrollarse en mas profundidad la parte del código del servidor. Con todas las partes definidas, la carpeta del *back-end* del proyecto quedaría así:

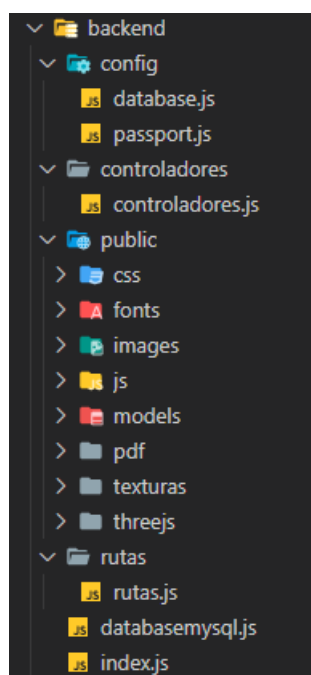


Figura 3.15: Back-End de la aplicación web.

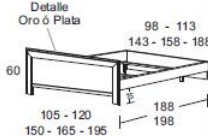


### 3.5.3. Diseño del Frontend

La idea para el diseño del *front-end* era que tuviese la esencia principal de las tiendas *online* de muebles, pero dándole un toque minimalista y moderno para que la experiencia del cliente fuese positiva y se mantuviese durante más tiempo en la página que, al fin y al cabo, sería el objetivo real de una tienda de muebles. Por lo tanto, la aplicación web dispondría de una página principal sencilla que tuviese los componentes necesarios para el desarrollo de la actividad.

Los dos elementos principales de la página web principal serían un catálogo para poder visualizar todos los muebles y acceder al configurador 3D y un acceso para los administradores al área privada.

Para el catálogo, se proporcionó un archivo *PDF* en el que se podía visualizar todos los muebles en imágenes junto a algunas de sus características como las dimensiones, código de referencia, etc. Para hacerse una mejor idea, el catálogo sería de la siguiente manera:

PARMA		ARO BAÑERA Nº 2			LIT Nº 2
DIMENSIÓN	REF.	DESCRIPCIÓN	COLOR	PUNTOS	
	624220 624221	Aro bañera piecero 90	180 190	213	
	624222 624223	Aro bañera piecero 105	180 190	230	
	624224 624225	Aro bañera piecero 135	180 190	Nogal Blanco	242
	624226 624227	Aro bañera piecero 150	180 190	247	
	624228 624229	Aro bañera piecero 180	180 190	261	
	624234 624235	Aro bañera patas 135	180 190	206	
	624236 624237	Aro bañera patas 150	180 190	Nogal Blanco	213
	624238 624239	Aro bañera patas 180	180 190	226	

9

Figura 3.16: Página del catálogo.

Una vez en posesión del catálogo en *PDF*, se debía buscar una manera para añadirle los enlaces al configurador 3D. Para que estos enlaces no estropeen el diseño del catálogo, lo

mejor era optar por poner los *links* en la referencia de cada mueble. El problema residía en que son muchas páginas del catálogo y muchos códigos de referencia por página, así que surgió la idea de convertir el archivos *PDF* a *HTML* y crear un *script* que convirtiese cada código en un enlace (el proceso se detalla en la sección 3.6).

Para el acceso al área privada de los administradores, debería abrirse una página para introducir el usuario y la contraseña y poder entrar a la página del dicho área. También se podría crear nuevos usuarios de administradores desde esta zona privada, por lo que también será necesario crear una página de registro.

Esto componentes serían los principales de las vistas de la aplicación, pero aun faltaría por especificar como será la página del configurador 3D y de que elementos contará. A continuación se abre un diagrama de navegación de la aplicación:

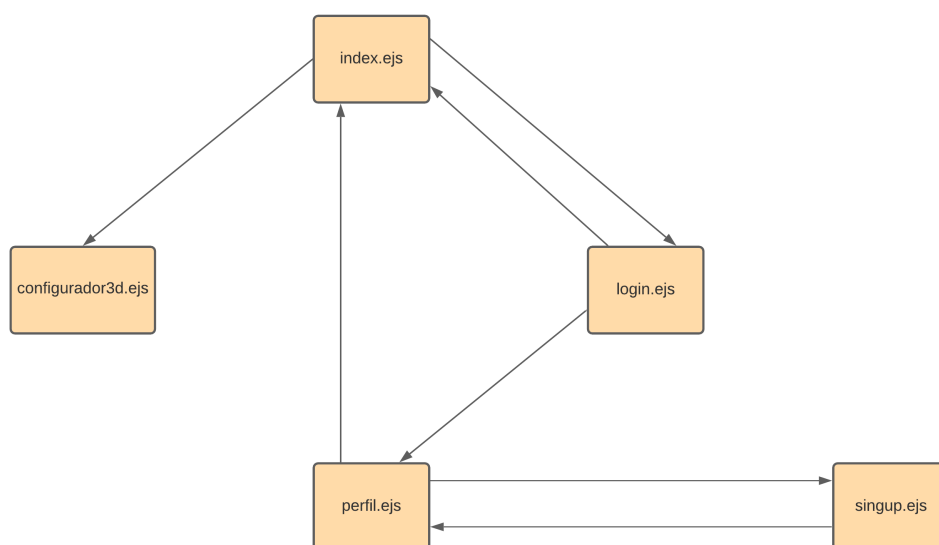


Figura 3.17: Diagrama de navegación.

Como se puede observar en el la figura 2.15, para poder registrarse como administrador en la aplicación, ha de ser otro el que lleve a cabo la operación, ya que si no cualquier usuario que ingresase a la aplicación podría crearse una cuenta de administrador.

### 3.5.4. Diseño del configurador 3D

El configurador 3D es la parte importante del proyecto y debería de estar reforzada con un buen diseño. Se requería un diseño simple y funcional, de modo que cualquier usuario que ingresase por primera vez al configurador pudiese usarlo sin necesidad de un manual, únicamente por intuición. Primero, se debía pensar cuales eran los componentes principales que debía tener la venta y se dictaminó que eran las texturas y el modelo en tres dimensiones del mueble seleccionado. Ambos elementos deberían aparecer en la misma pantalla para poder hacer uso del configurador con rapidez.

Se ha de mencionar que todas las texturas y todos los modelos han sido proporcionados al principio del proyecto y de los cuales, los modelos 3D son en formato *JS* y las texturas en formato *"PNG"*.

La tecnología elegida para este propósito mencionado en este diseño fue *Three.js*[1], la cual permite cargar el modelo, rotar y ampliarlo, y cambiar la textura señalando la parte deseada. Como todas las funciones que se iban a necesitar me las ofrecía esta librería, era perfecto para empezar el desarrollo e ir, poco a poco, cumpliendo con todos los requisitos definidos.

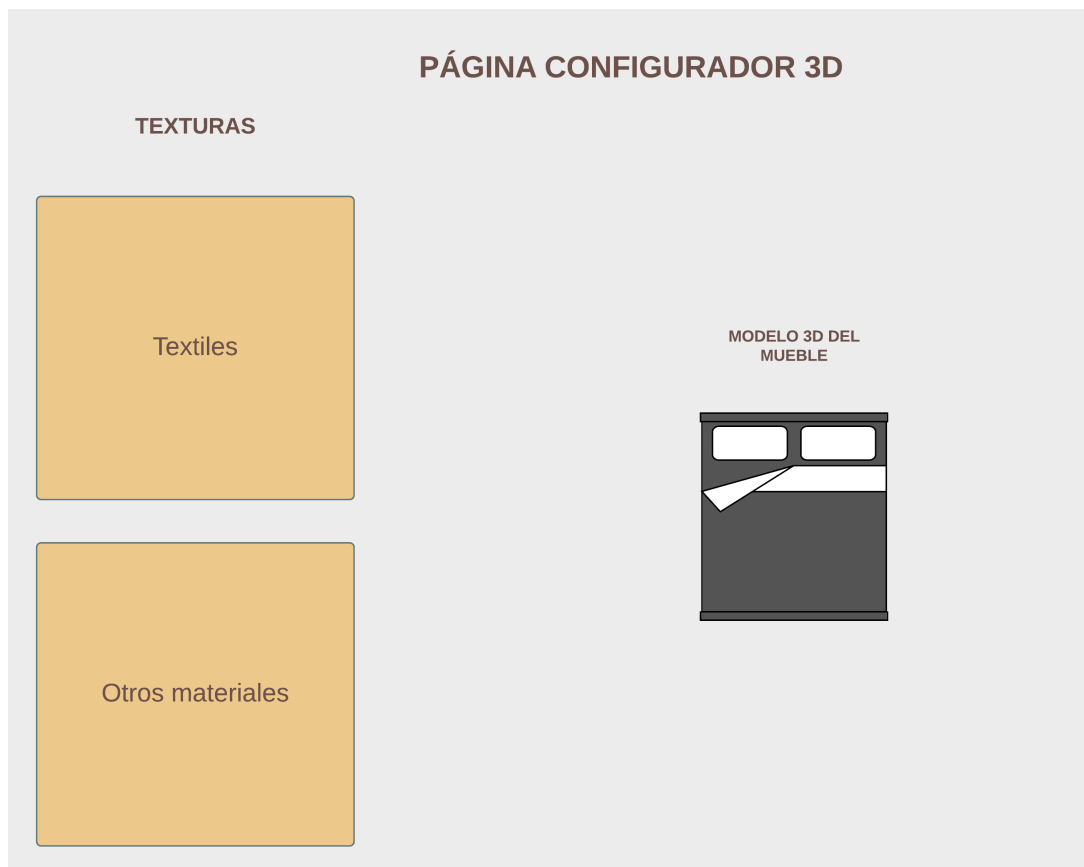


Figura 3.18: Diseño del configurador 3D.

## 3.6. Implementación

### 3.6.1. El catálogo

Como se ha mencionado anteriormente, se proporcionó un catálogo de muebles en formato en *pdf* y sería necesario modificar ese catálogo para añadir un *link* en cada código que viniese adjunto a cada mueble para que nos trasladase a la página del configurador 3D para dicho mueble. Tras pensar varias maneras de llevar a cabo esta tarea, se fueron probando varias tecnologías y librerías de JavaScript para automatizar este proceso, pero todas dieron malos resultados deformando el diseño del catálogo o colocando de manera errónea los códigos.

Tras varios fracasos se propuso la idea de convertir el catálogo a formato *HTML* y añadir los enlaces en los códigos correspondientes mediante *href*. Para ello se usó el conversor online *convertio.co* y se obtuvo con buenos resultado el catálogo en versión *web*.

El problema ahora residía en que el catálogo de los muebles es muy extenso y sería necesario encontrar una manera para automatizar el proceso de creación de enlaces en los códigos del catálogo. Es en este punto donde se plantean los dos objetivos principales para lograr este propósito: localizar los códigos en el catálogo y convertir estos en enlaces al configurador. La solución a estos problemas consistió en la creación de un *script* en *Python*, ya que es un lenguaje que nos permite hacer este tipo de funciones de manera rápida y simple.

Para localizar los códigos de los muebles se iteró por cada uno de los archivos *HTML* (cada uno se corresponde con una página del *PDF*) y se buscó en el código números que tuviesen como mínimo 6 caracteres, ya que los códigos pueden ser de 6 a 8 dígitos e ir acompañados de una letra "T". Teniendo esto en cuenta, una vez se localizan los códigos, se reemplaza la etiqueta *html* de texto por una de enlace a la dirección del configurador e indicando el código para cargar el modelo en tres dimensiones del mueble correspondiente.

Una vez se genera el catálogo con los enlaces, se vuelve a usar la herramienta de *convertio.co*[25] para revertir el formato, obteniendo el catálogo en *PDF* con los enlaces dispuesto para colocarlo en la página principal de la aplicación web. Para este propósito se utilizó un visor de archivos que proporciona la página *DocDroid*[8], que subiendo el catálogo a un repositorio y con un código con un *link* en *HTML* resolvía de manera muy sencilla el problema.

**PARMA** **ARO BAÑERA Nº 1** LIT Nº 1

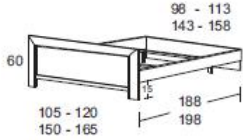
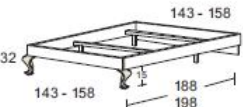
DIMENSIÓN	REF.	DESCRIPCIÓN	COLOR	PUNTOS
	624210 624211	Aro bañera piecero 90	180 190	185
	624212 624213	Aro bañera piecero 105	180 190	200
	624214 624215	Aro bañera piecero 135	180 190	215
	624216 624217	Aro bañera piecero 150	180 190	225
	624234 624235	Aro bañera patas 135	180 190	206
	624236 624237	Aro bañera patas 150	180 190	213
			Nogal	
			Blanco	
 <p>Color patas: Dorado - Plata - Antracita</p>				

Figura 3.19: Localización de los códigos.

Como se puede observar en la *Figura 2.17*, cada mueble del catálogo tiene asociado un código que corresponde con el modelo en 3D que se debe cargar en el configurador. Sabiendo como es el diseño de este catálogo se procederá al proceso mencionado anteriormente para generar el mismo catálogo pero con los enlaces añadidos.

### 3.6.2. La base de datos

La implementación de la base de datos fue sencilla, ya que únicamente se tuvo que añadir dos tablas para almacenar datos que se usan en nuevas implementaciones en la aplicación web.

La primera tabla que se añadió fue *Users*, que almacena datos personales de los administradores para que estos pudiesen iniciar sesión y acceder al área privada y ver algunos datos personales. La estructura de la tabla es la siguiente:

Administradores
idUsuario :: int (PK)
usuario :: varchar
contraseña :: char
nombreCompleto :: varchar
edad :: int
puesto :: varchar

Figura 3.20: Tabla de usuarios de la base de datos.

La segunda tabla se almacenan datos del encargo realizado por el cliente. También se almacena el nombre del fichero de la imagen del mueble ya personalizado para que los fabricantes de la tienda tengan claro como es el mueble deseado por el cliente. Estos datos se muestran en el área privada de los administradores. La tabla tiene las siguientes columnas

Administradores
idUsuario :: int (PK)
usuario :: varchar
contraseña :: char
nombreCompleto :: varchar
edad :: int
puesto :: varchar

Figura 3.21: Tabla de petición de encargos de muebles.

Se ha de mencionar que la herramienta que se ha usado para la administración de la base de datos ha sido *HeidiSQL*[6], que ha permitido gestionar de manera muy cómoda e intuitiva la base de datos creada en *MariaDB*[5].

El resto de tablas y relaciones de la base de datos son usados en la parte del servidor de la aplicación y son esenciales para el funcionamiento de la aplicación. Más adelante en la memoria se explicará como se han utilizado todos estos datos y tablas para obtener la información necesaria para la ejecución de la aplicación. Se va a mostrar a continuación un listado de las diferentes tablas con los datos que almacenan y sus relaciones con otras tablas:

- **Piezas:** contiene información sobre la pieza (mueble) tal como el nombre, tipo, medidas, precio, etc. Respecto a la relaciones con otras tablas, pertenece a *Subcategorías*, permite diferentes *Texturas* y pertenece a *GrupoPieza*.

- **Subcategoría:** tabla intermedia entre *Categorías* y *Piezas*, aunque esta primera no se ha añadido al proyecto porque no era necesaria. Esta tabla también posee una breve descripción de la subcategoría. Se relaciona con *Piezas*.
- **GrupoPieza:** esta tabla indica que las piezas se agrupan en grupos y la información que contiene es respecto al id del fabricante, la venta, descripción y tarificación. Se relaciona con la tabla *Piezas* y la tabla *Fabricantes*.
- **Fabricantes:** contiene información de los fabricantes y se relaciona con la tabla *GrupoPieza*.
- **Texturas:** almacena información sobre las texturas que se pueden aplicar a las piezas. Se relaciona con *Piezas* indicando cuales de esas texturas están disponibles, con *PartePiezas* informando que texturas se pueden aplicar a cada parte de cada pieza y con la tabla *TipoTexturas*.
- **PartePiezas:** la información de esta tabla representa las subpartes que componen a la pieza completa. Se relaciona con la tabla *Texturas*.
- **TipoTexturas:** contiene información adicional de las texturas y es por ellos que se relaciona con dicha tabla.

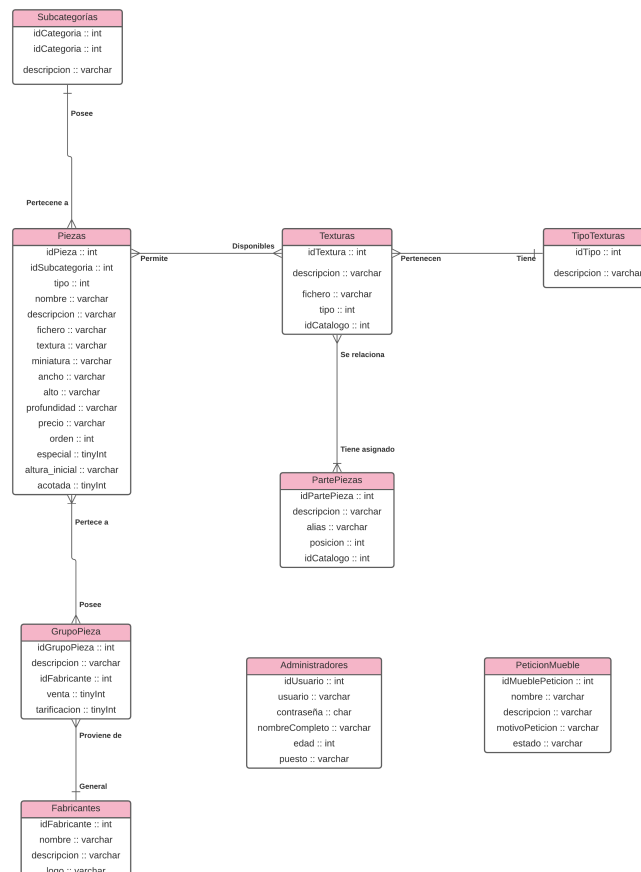


Figura 3.22: Diagrama Entidad-Relación de la base de datos.

### 3.6.3. Implementación del Front-End

Para la implementación del *Front-End* se ha respetado estrictamente el diseño del proyecto y el resultado ha consistido en los siguientes componentes y vistas:

- **Página principal:** La página principal necesitaba un buen diseño ya que es lo primero que los clientes de la tienda van a percibir y es necesario que estos se sientan a gusto con la página. Un punto importante es la simplicidad de la página para que los usuarios de manera intuitiva puedan ser capaces de hacer un uso correcto y no necesitar de un manual o una consulta para ello. Toda la implementación de esta página se encuentra en el fichero *index.ejs* y sus componentes más importantes son:
  - Un visor del catálogo que se ha mencionado anteriormente en *PDF* disponible para ver o descargar.
  - Un formulario para que el cliente pueda encargar un mueble personalizado por ellos mismos. Los elementos de este formulario son el nombre del mueble, una descripción por si fuese necesario especificar algo del mueble, un motivo para comprarlo (si se quiere alquilar o comprar), estado (recién fabricado o reutilizado) y la imagen del mueble personalizado. Llama al método POST */encargo*.
  - Un enlace para iniciar sesión y entrar al área privada de los trabajadores.
  - Adicionalmente se ha añadido un enlace que nos traslada al perfil de la red social *LinkedIn* del creador de del proyecto.

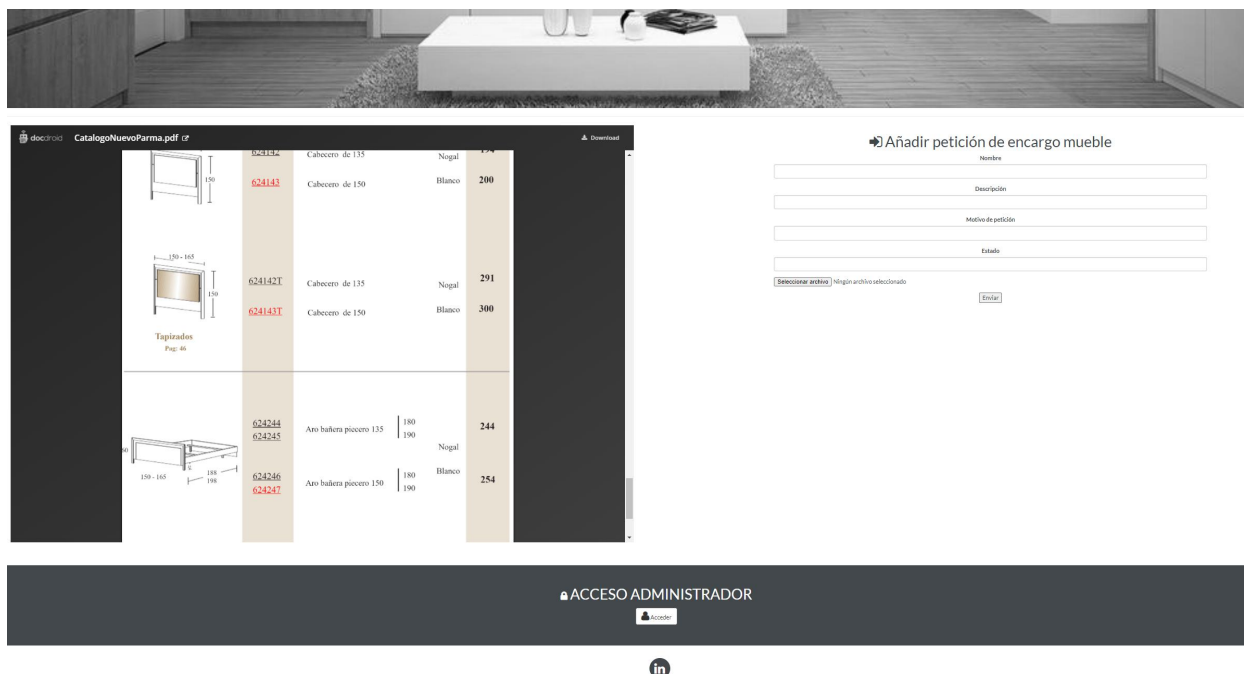


Figura 3.23: Vista página principal.

- **Configurador 3D:** La vista del configurador en tres dimensiones de los muebles se accede desde el catálogo de los muebles, el fichero se llama *configurador3D.ejs* y consta de tres elementos:



- Dos tablas, una con las diferentes texturas textiles disponibles para el mueble y otra con el resto de materiales.
- El modelo en 3D del mueble el cual se puede ampliar, rotar y seleccionar elementos para aplicar las texturas.
- Un botón para descargar una imagen del mueble ya personalizado.

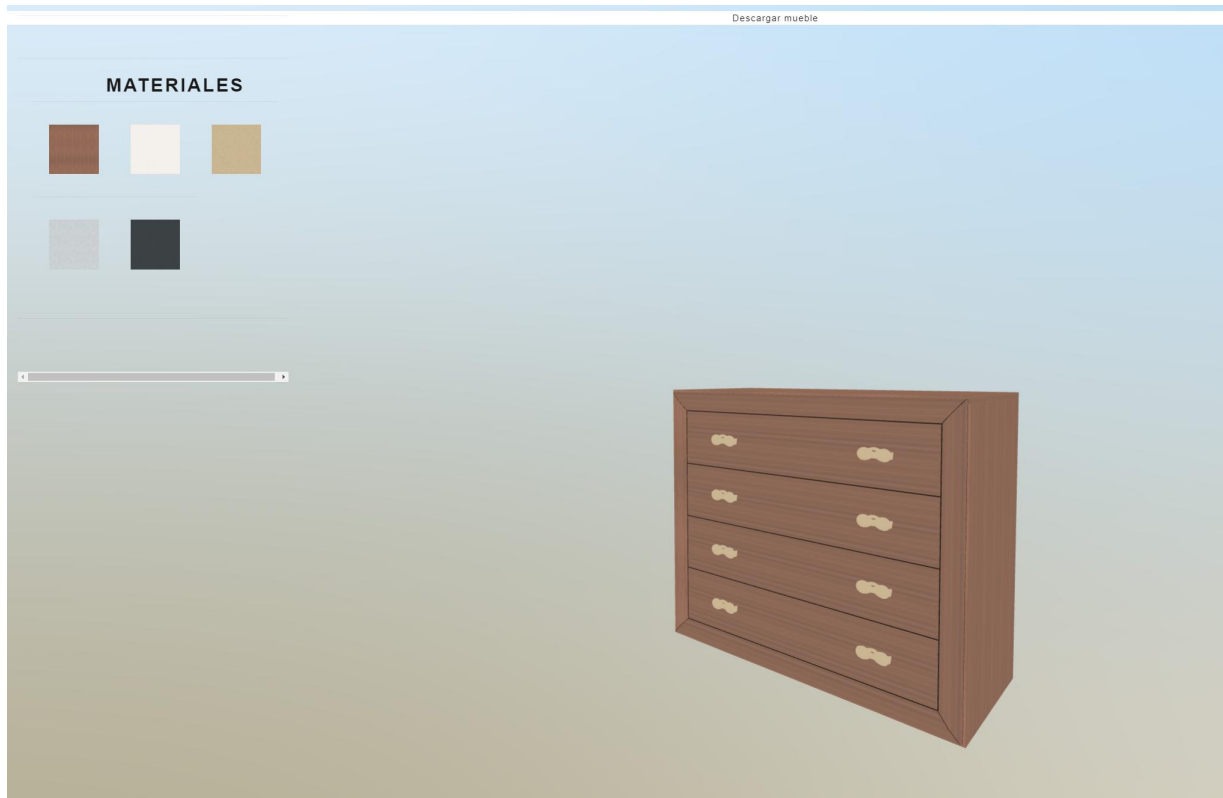
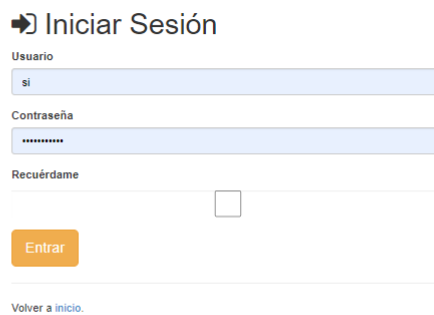


Figura 3.24: Vista del configurador 3D.

- **Ventana de inicio de sesión:** Contiene un formulario para introducir un usuario y la contraseña y acceder al área privada. Tiene un *checkbox* con motivo de no tener que escribir estos campos cada vez que se quiera acceder. También dispone de un enlace para volver a la página principal y el botón de *submit* que llama al método POST de */login*. El fichero que implementa esta vista se llama *login.ejs*.
- **Área privada de administradores:** La implementación de esta ventana se lleva a cabo en el fichero *perfil.ejs*. Contiene dos tablas, la primera muestra la información personal del administrador que accede (nombre de usuario, nombre completo, edad y puesto de trabajo). La segunda tabla muestra los encargos realizados por los clientes y permite eliminarlos mediante un botón que lleva al método */eliminar* y contiene los siguientes elementos: nombre del mueble, descripción, motivo de petición, estado y una imagen del mueble. También permite mediante un botón trasladarnos a la ventana de registro de nuevo usuario, otro para cerrar sesión y un último para volver a la página principal.
- **Ventana de registro de usuario:** La implementación de esta vista se encuentra en *signup.ejs*. Se puede introducir los siguientes datos: usuario, contraseña, nombre completo, edad y puesto de trabajo. El botón de *submit* *Registrar* lanza el formulario



➡ Iniciar Sesión

Usuario  
si

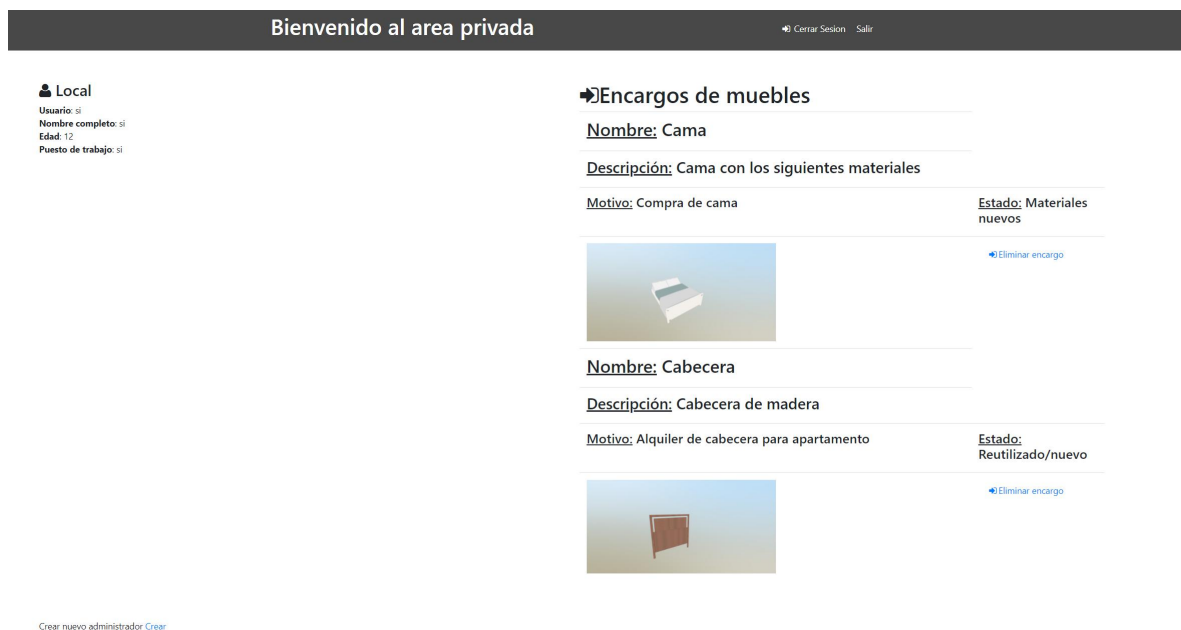
Contraseña  
\*\*\*\*\*

Recuérdame ☐

Entrar

[Volver a inicio.](#)

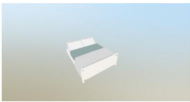

Figura 3.25: Vista de la página de inicio de sesión.



Bienvenido al area privada [Cerrar Sesión](#) [Salir](#)

**Local**  
Usuario: si  
Nombre completo: si  
Edad: 12  
Puesto de trabajo: si

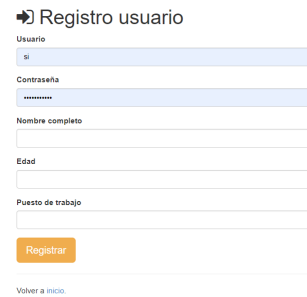
**➡ Encargos de muebles**

<b>Nombre:</b> Cama	
<b>Descripción:</b> Cama con los siguientes materiales	
<b>Motivo:</b> Compra de cama	<b>Estado:</b> Materiales nuevos
	<a href="#">Eliminar encargo</a>
<b>Nombre:</b> Cabecera	
<b>Descripción:</b> Cabecera de madera	
<b>Motivo:</b> Alquiler de cabecera para apartamento	<b>Estado:</b> Reutilizado/nuevo
	<a href="#">Eliminar encargo</a>

[Crear nuevo administrador](#) [Crear](#)

Figura 3.26: Vista del área privada de los administradores.

ejecutando el método *POST /signup*. Esta ventana también contiene un enlace para volver a la página de inicio.



Registro usuario

Usuario

SI

Contraseña

.....

Nombre completo

Edad

Puesto de trabajo

Registrar

[Volver a inicio](#)

Figura 3.27: Vista de la pantalla de registro de nuevo usuario.

También es importante destacar el *view engine* utilizado para todas las vistas del proyecto, el cual es *ejs*. Este motor de vistas nos permite mantener todas las funcionalidades de los ficheros *html*, pero adicionalmente nos permite introducir código en JavaScript entre sentencias y etiquetas de *html*, lo cual ofrece mucha libertad y dinamismo en las vistas ya que nos permite funcionalidades como iterar y crear tablas dinámicas o acciones de control.

A continuación se muestra una imagen del directorio que contiene todos los archivos del *frontend* que conforman las vistas de la aplicación.

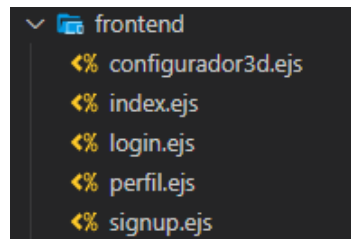


Figura 3.28: Directorio del *frontend* del proyecto.

### 3.6.4. Implementación del Back-End

En un enfoque general, la implementación del *Backend* ha consistido en la creación de una *API Rest* que contiene todos los métodos necesarios para que haya un flujo de datos entre la base de datos y la parte del cliente. Estos métodos y el servidor en general constituyen el esqueleto de la aplicación y es necesario que tenga un funcionamiento correcto para que la aplicación cumpla con todas sus funciones.

Sin duda una de las partes más importantes y complejas de este proyecto, junto al configurador 3D, ha sido la implementación de la parte del *backend* de la aplicación. Para ello inicialmente se ha creado la estructura de una aplicación de *Node.js*[26] y *Express*[2] típica para ir desarrollándola y añadiendo funcionalidades de manera iterativa. A continuación se va a mostrar una lista de los componentes y funcionalidades más importantes que se han implementado en la aplicación web y con que herramientas o funciones se han hecho:

- Para la conexión a la base de datos se ha usado el *plug-in* de *Node mysql* y se ha declarado todos los parámetros necesarios para la conexión a mi base de datos. Esto es un proceso muy sencillo y ya se podría acceder a los datos sin ninguna complicación. El archivo donde se configura esta conexión se ha llamado *databasemysql.js*.
- Se ha creado un directorio estático llamado *public* para almacenar todos aquellos archivos necesarios en la parte del servidor. Se ha incluido subcarpetas dentro de este directorio para desarrollar el proyecto de manera más ordenada y entre estas existirían carpetas para guardar los archivos *CSS*, *JS*, imágenes, modelos, librerías, texturas, etc. La definición de este directorio se encuentra en el archivo de configuración principal de *Node*, al que he llamado *index.js*.
- Se ha creado un archivo de rutas llamado *rutas.js* donde se han definido todos los métodos de la *API Rest* y las rutas en sí mismas de la aplicación web para hacer posible la navegación y el envío y recepción de datos. Los métodos más destacables son los siguientes:
  - Para el sistema de inicio de sesión, se han empleado diversos métodos *GET* y *POST* que usan dos *plug-ins* de *Node* llamados *Bcrypt* y *Passportjs*. El primero mencionado es una función *hash* de cifrado usado para las contraseñas de inicio de sesión, el segundo es usado para facilitar el sistema de login y accesos a la base de datos. La configuración de *Passportjs* se encuentra en el fichero *passport.js* y usa un fichero adicional de configuración de base de datos que se llama *database.js*.
  - Al igual que hay un sistema de inicio de sesión, también hay otro de registro, que en resumidas cuentas, es muy similar al de login (también usa *Passport* y *Bcrypt*), pero en vez de consultar en base de datos, lo que hace es introducirlos añadiendo así nuevos usuarios.
  - Uno de los métodos más importantes es el método *GET* que lleva a la ruta del configurador 3D. Debido a la extensión de este método, se ha creado un nuevo fichero llamado *controladores.js* en el que se ha guardado todo el contenido del mismo. Este método destaca por contener cinco consultas a la base de datos anidadas cuya función ha sido obtener todos los datos necesarios para enviar al configurador 3D. Han sido necesarias tantas consultas debido al gran número de relaciones y tablas de la base de datos y también debido al alto filtrado que

había que hacer de los datos. Los datos que se envían al configurador son las texturas para el modelo completo, las texturas disponibles para cada elemento del modelo y el nombre del modelo junto a otros datos para poder cargarlo.

- Otros métodos relevantes son los que insertan en la base de datos la información del formulario del encargo en la página principal (*POST*) y los que permiten mostrar en el área privada de los administradores la información personal, mostrar los encargos realizados por los clientes y el que permite eliminar dichos encargos (todos también *POST*).

Durante el desarrollo del *Backend* también se han usado otros *plug-ins* que ayudan al desarrollo como *Mongoose*, que permite captar los cambios de la aplicación en tiempo real y *multer* para cargar ficheros desde las vistas.

A continuación se muestra una imagen del directorio del proyecto donde se observa la estructura del *backend* con todos sus ficheros.

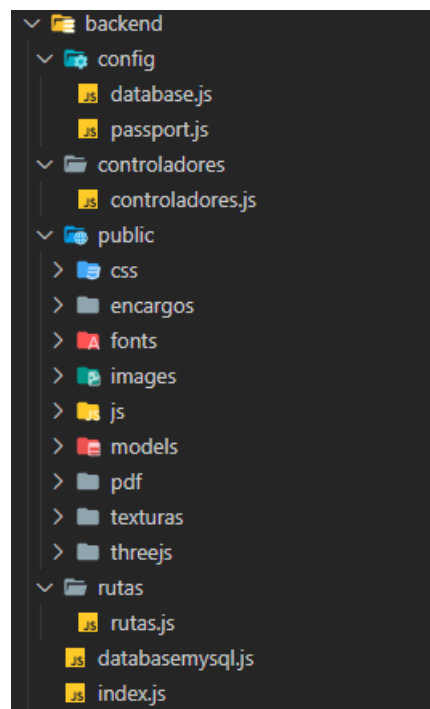


Figura 3.29: Directorio del *backend* del proyecto.

### 3.6.5. Implementación del configurador 3D

La implementación del configurador era la más importante de este proyecto y su funcionamiento debía ser correcto y cumplir con el diseño mencionado anteriormente. En la vista del configurador se pueden encontrar tres elementos principales: una tabla con las texturas disponibles para el modelo 3D del mueble, el modelo del mueble en 3D y un botón para descargar una imagen del modelo en 3D.

Para esta vista se ha usado *Three.js*[27] y sus componentes para convertir una página web en 2D a una que incluye elementos en 3D y que permite una serie de funcionalidades que la convierten en una opción muy dinámica y atractiva para el cliente. A continuación, se muestra una lista con los métodos usados de *Three.js* y la función que desempeñan en la aplicación:

- **Scene():[28]** Crea la escena, la cual es imprescindible para tener un entorno para añadir los distintos elementos del configurador. Es donde se cargan el resto de componentes de *Three.js*.
- **PerspectiveCamera():[29]** Coloca en el escenario una cámara con perspectiva (imita la vista humana). Es lo que permite al usuario visualizar en la pantalla la escena creada y cada uno de sus componentes.

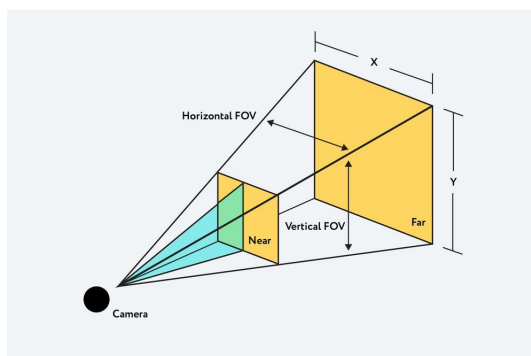


Figura 3.30: Cámara con perspectiva.

- **WebGLRenderer():[30]** Se encarga de renderizar la escena con todos sus componentes para que se pueda cargar en la página web. La escena se está renderizando constantemente en bucle para actualizarse ante cualquier cambio en el entorno.
- **PointLight():[31]** Añade un punto de luz en la localización deseada y se define también el color de esta. Sin luz la escena se vería en negro, por lo que es imprescindible.
- **TextureLoader():[32]** A partir de una imagen, crea una textura dispuesta a convertirse en un material. En el caso de esta aplicación se usa para cargar las diferentes texturas disponibles para el mueble seleccionado.
- **MeshBasicMaterial():[33]** Método que convierte las texturas en materiales para que se puedan aplicar a los modelos 3D. En este caso carga las texturas generadas y las convierte en materiales para los muebles.
- **LegacyJSONLoader():[34]** Método que se usa para cargar los modelos de los muebles con extensión *.js*. Para cargarlo necesita de materiales y la geometría del

modelo en si mismo.

- **OrbitControl()**[35]: Permite desplazar la cámara con libertad y rotarla. En este caso se ha añadido para que se pueda apreciar de la mejor manera todo el mueble.
- **RayCaster()**[36]: Método que detecta modelos en 3D dentro de la escena pinchando en la vista de la aplicación. La he usado para detectar los componentes de cada modelo y para aplicar los materiales a determinadas partes del mueble.



Figura 3.31: Respresentación RayCast.

Como se observa en la figura 2.28, el método *RayCaster()* manda un “rayo” que sale desde la cámara y atraviesa todos los objetos de la escena que se encuentren en su camino y almacena en un *array* información de todas las caras de objetos detectadas.

Todos estos métodos se definen dentro de la función *ini()*, que inicializa la escena con todos sus componentes. Luego la función *animate()* se encarga de renderizar la escena en bucle y actualizar la posición de la cámara según el usuario vaya manipulando la escena desde la vista.

Encajando cada uno de estos métodos y especificando parámetros dentro del código, el resultado final del configurador en 3D es el siguiente:

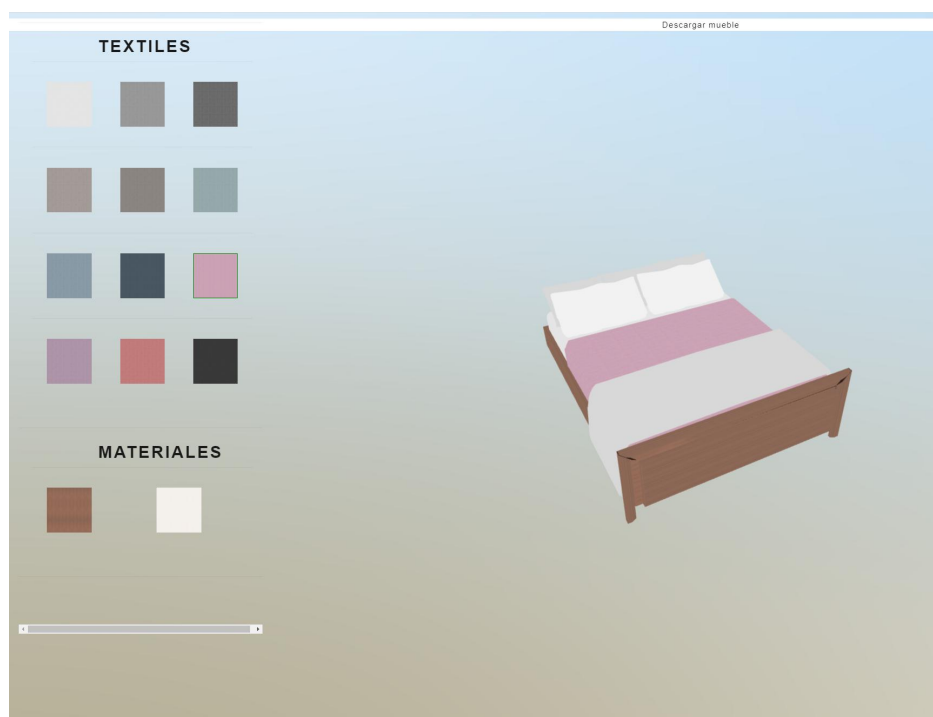


Figura 3.32: Vista de la ventana del configurador 3D.

### 3.7. Pruebas

En esta sección se van a detallar una serie de pruebas para evidenciar algunas partes de la implementación del proyecto, ya que hay ciertos aspectos en los que, aunque se han mencionado en la parte de la implementación, no se ha presentado ninguna prueba o evidencia de su funcionamiento.

#### 3.7.1. Caso de prueba 1

- **Título:** Aplicar textura permitida.
- **Descripción:** Se procede a aplicar una textura en el configurador 3D a un mueble en una zona en la cual se permite la aplicación de dicha textura.
- **Evidencia:**

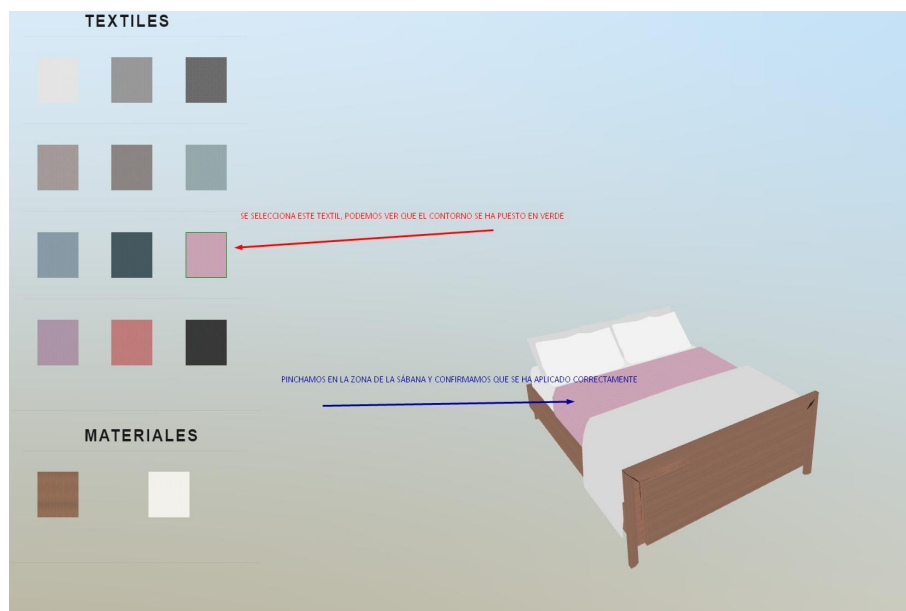


Figura 3.33: Evidencia del caso de prueba 1.

#### 3.7.2. Caso de prueba 2

- **Título:** Aplicar textura no permitida.
- **Descripción:** Se procede a aplicar una textura en el configurador 3D a un mueble en una zona en la cual no se permite la aplicación de dicha textura.
- **Evidencia:**





Figura 3.34: Evidencia del caso de prueba 2.

### 3.7.3. Caso de prueba 3

- **Título:** Inicio de sesión con credenciales correctas.
- **Descripción:** Se inicia sesión con un usuario y una contraseña previamente registrados. Tras hacerlo se accederá al área privada.
- **Evidencia:**



Figura 3.35: Evidencia del caso de prueba 3 (parte 1).

### 3.7.4. Caso de prueba 4

- **Título:** Inicio de sesión con credenciales incorrectas.

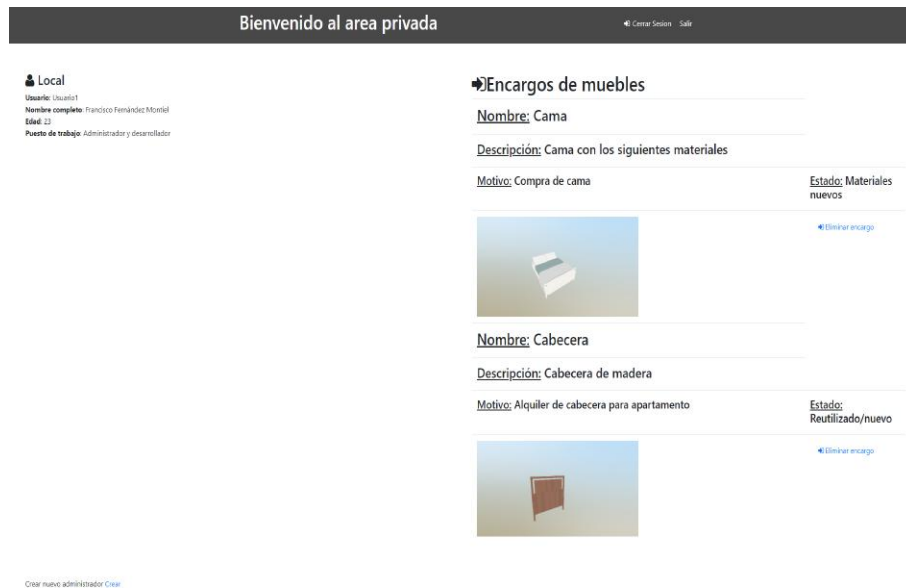


Figura 3.36: Evidencia del caso de prueba 3 (parte 2).

- **Descripción:** Se inicia sesión con un usuario y una contraseña que no han sido registrados previamente. Tras hacerlo no se podrá acceder al área privada.
- **Evidencia:**

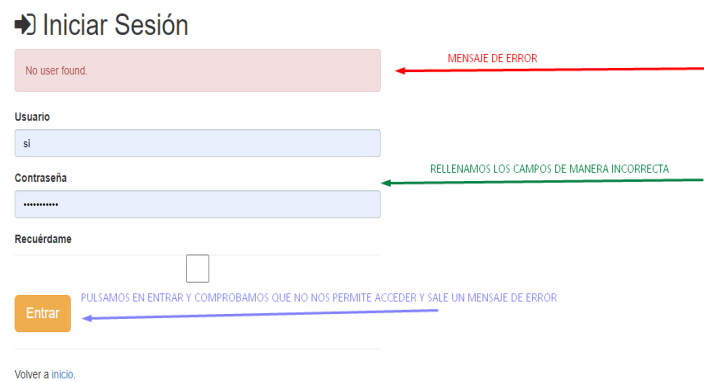


Figura 3.37: Evidencia del caso de prueba 4.

# CAPÍTULO 4

---

## Conclusiones

---

### 4.1. Conclusiones técnicas y aprendizaje personal

Durante el desarrollo de este proyecto ha sido necesario formarme en el uso de nuevas tecnología desconocidas hasta el momento, así como profundizar en aquellas que había rozado de manera superficial.

Entre las tecnologías utilizadas podemos destacar el avanzado nivel de *Latex* que he adquirido, tecnología que solamente había utilizado en contadas ocasiones de la carrera y extremadamente útil para la generación de textos académicos como esta misma memoria.

Se debe mencionar el progreso personal que he tenido con la tecnología que me hizo decantarme por esta línea de trabajo y esa tecnología es *Threejs*. De crear escenas simples con iluminación a ser capaz de importar todo tipo de modelos en tres dimensiones, poder realizar movimientos con la cámara situada en la escena de rotación y desplazamiento, usar herramientas complejas como el *RayCast* para detectar objetos en la escena, añadir animación a los objetos y muchas más funcionalidades que me han entusiasmado y que han provocado que quiera aprender más sobre esta tecnología y llevar a cabo nuevos proyectos en un futuro.

También destacaría de mi aprendizaje personal la nueva capacidad que he adquirido de crear en aplicaciones web controladores más complejos para obtener información de bases de datos muy elaboradas y con muchas relaciones y ser capaz de filtrar esta información y manipular los datos a mi gusto para enviarlos a las vistas de la aplicación y usarlas con plena libertad y sin restricciones debido al desconocimiento.

Aunque es cierto que ya había realizado algún proyecto personal simple, mis conocimientos en creación de páginas web con *Nodejs* y *Express* era muy limitado, es por ello que una gran parte del reto de este trabajo ha sido mejorar mis habilidades con estas tecnologías e ir descubriendo que herramientas y *plug-ins* disponían para ayudar a la programación de la aplicación web.

Por último, me gustaría mencionar que gracias a este proyecto, he podido conocerme mejor a mi mismo incluyendo debilidades y fortalezas, conocer mejor mis capacidades y habilidad con la informática y aprender que con interés y dedicando tiempo la programación web no es un área difícil y que incluso si lo enfocas de la manera correcta puede resultar bastante entretenido.

### 4.2. Dificultades encontradas

En esta sección se tratarán de abarcar las dificultades más significativos encontrados durante el desarrollo de este proyecto, donde al ser tan extenso y con tantos frentes abiertos, han sido bastante frecuentes.

Uno de los primeros problemas a los que me enfrenté, fue a la hora de utilizar los controladores para obtener los datos de la base de datos utilizada, puesto que la misma es muy compleja, al tener muchas relaciones entre modelos texturas y las distintas filas.

Otro de los problemas que surgieron a la hora de cargar los modelos 3D proporcionados al inicio del proyecto, fue que los mismos eran archivos de extensión *.js*, los cuales no se acercan a lo que serían ficheros de modelos en tres dimensiones estándar y que, por lo cual, resultaba complicado la implementación de los mismos en nuestra aplicación debido a la incompatibilidad de estos para cargarlos en mi herramienta de desarrollo. Tras investigar en profundidad, fui capaz de solucionar este problema descargando versiones más antiguas de *ThreeJS* que incluían unos módulos adicionales que sí permitían la carga de este tipo de archivos.

Adicionalmente me gustaría destacar la especial dificultad de subdividir un modelo en distintas partes para poder así aplicar diferentes texturas al mismo. Ya que aunque aplicar una textura a un modelo 3D es algo bastante simple, cuando por ejemplo queremos aplicar varias texturas distintas a por ejemplo, una cama, donde el canapé el colchón y las sábanas están formados por distintos materiales la tarea se nos dificulta enormemente. Esto, tras investigar de nuevo pude solventarlo utilizando un componente de *ThreeJS* llamado *RayCast*.

El último problema que me gustaría mencionar es mi propia responsabilidad y capacidad para ser capaz de priorizar este proyecto sobre otros en los que me encontraba involucrados. No acabar la carrera y encontrar un trabajo te crea una impresión falsa de haberla terminado cuando, en realidad, no es así. Este hecho provocó que le dejase de dar importancia a este proyecto y cada vez se me hacía más cuesta arriba finalizarlo. Finalmente fui capaz de madurar y reflexioné sobre el tema, consiguiendo la motivación que me hacía falta para terminar el proyecto y, lo más importante, disfrutar todo el proceso.

### 4.3. Futuras líneas de trabajo

Este proyecto al tratarse de una aplicación web, podría expandirse en un futuro a una plataforma de comercio electrónico, añadiendo algún método de pago seguro con el que poder solicitar los muebles deseados con una previsualización previa.

También se podría actualizar la tecnología 3D y asignar una mayor cantidad de recursos para poder así generar modelos 3D de una calidad superior, quizás incluso usando tecnología que otorgue resultados fotorrealistas, aplicando técnicas como el *Raytracing*.

Otra de las posibilidades sería desarrollar una versión para móvil de la página, e incluso una aplicación local donde a través del uso de realidad aumentada poder visualizar como quedarían los muebles personalizados en la habitación antes de comprarlos.

También si deseáramos utilizar esta página web en un entorno comercial, podríamos aumentar la seguridad de los protocolos utilizados, añadir más funcionalidades al administrador para otorgarle un mejor control de la página. Y finalmente realizar un pulido del diseño general del *frontend*, para ofrecerle al cliente una página web aún más atractiva, intuitiva y agradable a la vista.



---

## Bibliografía

---

1. *three.js*, (<https://threejs.org/docs/>).
2. *Infraestructura de aplicaciones web Node.js*, (<https://expressjs.com/es/>).
3. *bcrypt*, (<https://www.npmjs.com/package/bcrypt>).
4. *Documentation*, (<http://www.passportjs.org/docs/>).
5. *MariaDB Server 10.6.3*, (<https://mariadb.com/kb/en/mariadb-server-10-6-3/>).
6. A. Becker, *Download HeidiSQL 11.3, released on 30 May 2021*, (<https://www.heidisql.com/download.php>).
7. (<https://getbootstrap.com/docs/5.1/>).
8. *Upload PDF*, (<https://www.docdroid.net/>).
9. (<https://fontawesome.com/v5.15/how-to-use/on-the-web/referencing-icons/basic-use>).
10. J. F. js.foundation, *jQuery API*, (<https://api.jquery.com/>).
11. Microsoft, *Visual Studio Code - Code Editing. Redefined*, abr. de 2016, (<https://code.visualstudio.com/>).
12. *What is Notepad*, (<https://notepad-plus-plus.org/>).
13. *Online latex editor*, (<https://www.overleaf.com/>).
14. (<https://calendar.google.com/>).
15. (<https://git-scm.com/>).
16. *Where the world builds software*, (<https://github.com/>).
17. *Software de diagramación en línea y solución visual*, (<https://www.lucidchart.com/pages/es>).
18. J. Diego Gauchat, *El gran libro de HTML5, CSS3 y JavaScript 3ª Edición* (MARCOMBO S.A, 2019).
19. *Welcome to Python.org*, (<https://www.python.org/>).
20. *CONFIGURATORE 3D*, (<https://configurator.lago.it/>).
21. (<https://decotreku.treku.es/Treku/>).
22. W. P. s.r.l., *mesas*, (<https://www.cattelanitalia.com/es/products/configurator?c=2>).
23. (<https://lanmobel.online3dplanner.com/planificador/>).

24. J. Dickey, *Write modern web apps with the MEAN stack: Mongo, Express, AngularJS, and Node.js* (Peachpit Press, 2015).
25. *Convertidor de archivos*, (<https://convertio.co/es/>).
26. Node.js, (<https://nodejs.org/es/>).
27. D. Jos, *Learning Three.js – the JavaScript 3D Library for WebGL - Second Edition*. (Packt Publishing Ltd, 2015).
28. *Three.js - Creating a scene*, (<https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene>).
29. *Three.js - PerspectiveCamera*, (<https://threejs.org/docs/index.html#api/en/cameras/PerspectiveCamera>).
30. *Three.js - WebGLRenderer*, (<https://threejs.org/docs/index.html#api/en/renderers/WebGLRenderer>).
31. *Three.js - PointLight*, (<https://threejs.org/docs/index.html#api/en/lights/PointLight>).
32. *three.js - TextureLoader*, (<https://threejs.org/docs/index.html#api/en/loaders/TextureLoader>).
33. *Three.js - MeshBasicMaterial*, (<https://threejs.org/docs/index.html#api/en/materials/MeshBasicMaterial>).
34. *Three-legacyjsonloader: Docs, Tutorials, Reviews*, (<https://openbase.com/js/three-legacyjsonloader>).
35. *Three.js - OrbitControls*, (<https://threejs.org/docs/index.html#examples/en/controls/OrbitControls>).
36. *Three.js - Raycaster*, (<https://threejs.org/docs/index.html#api/en/core/Raycaster>).



## Apéndice



# APÉNDICE A

## Manual de Instalación

### A.1. Requisitos

- Disponer un ordenador de características mínimas que permita la instalación de los componentes mencionados a continuación.
- Tener el código fuente descargado en nuestro ordenador.
- Tener descargado alguna herramienta de descompresión de archivos con extensión *.zip*. Recomendado *WinRaR* versión 6.02.



Figura A.1: Página de descarga de WinRaR.

- Instalar en el caso que no se tenga el entorno de ejecución *Node.js* versión 14.17.6.



Figura A.2: Página de descarga de Node.js.

- Dado que el archivo comprimido ocupa 173 MB y el código fuente descomprimido 298 MB, necesitaremos 471 MB libres en nuestra computadora.
- Por último, necesitaremos un navegador web para visualizar la aplicación una vez se compile. Funciona en los principales navegadores, pero se recomienda *Mozilla Firefox* o *Google Chrome* (últimas versiones).

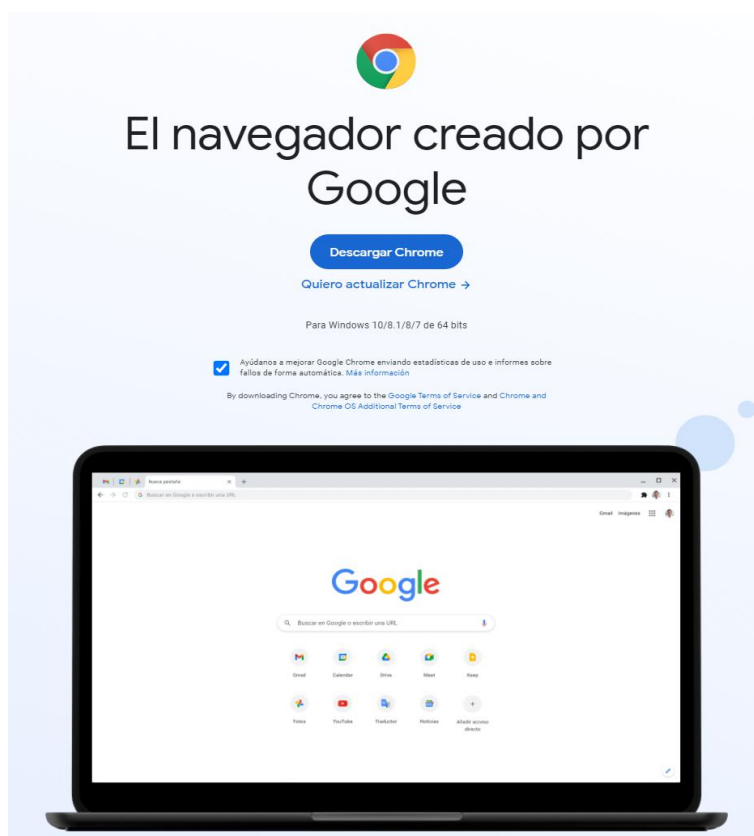


Figura A.3: Página de descarga de Google Chrome.



Figura A.4: Página de descarga de Mozilla Firefox.

## A.2. Paso para la instalación

- Se descomprime el código fuente con la herramienta de descompresión.
- Se abre el terminal de la computadora y mediante el comando `“cd”` nos situamos dentro de la carpeta que hemos descomprimido indicando la ruta.
- Se ejecuta en el terminal el comando `“npm install”` para instalar las dependencias del proyecto.
- Se ejecuta el comando `“npm run start”` para compilar el proyecto.
- Por último, dentro de la barra de búsqueda del navegador introducimos la siguiente URL: `“http://localhost:3000/”`.



# APÉNDICE B

## Manual de Usuario

En este apéndice se va a exponer un manual de uso para el cliente y otro para el administrador. En el caso del cliente, se va a explicar, desde la página de inicio, como personalizar un mueble del catálogo y realizar un encargo. Como administrador se va a indicar como acceder al área privada, visualizar los datos y registrar un nuevo administrador.

### B.1. Manual cliente

- Desplazarse hasta el catálogo en la página principal y elegir un mueble. Para acceder al configurador se deberá pinchar en el código asociado al mueble.

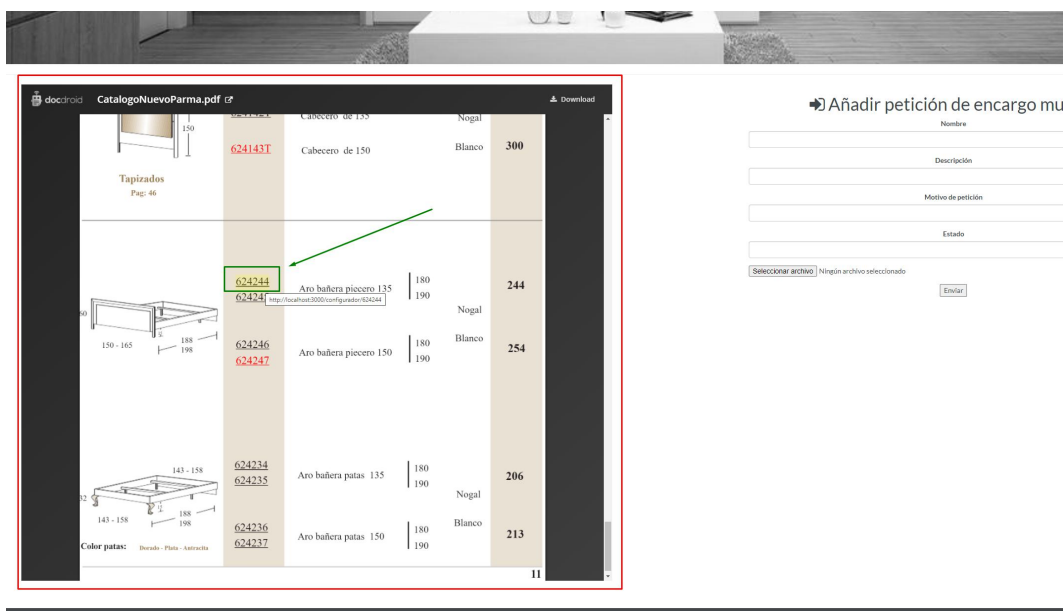


Figura B.1: Página principal de la aplicación (catálogo).

- En esta ventana se podrá ampliar y rotar el mueble al gusto pinchando en la pantalla y con la rueda del ratón. Para cambiar las texturas se deberá pinchar en la deseada en la tabla de la izquierda y hacer *click* en la parte del mueble donde se quiere aplicar.
- Una vez personalizado el mueble se podrá descargar una imagen del mueble pinchando en el botón situado en la parte superior.

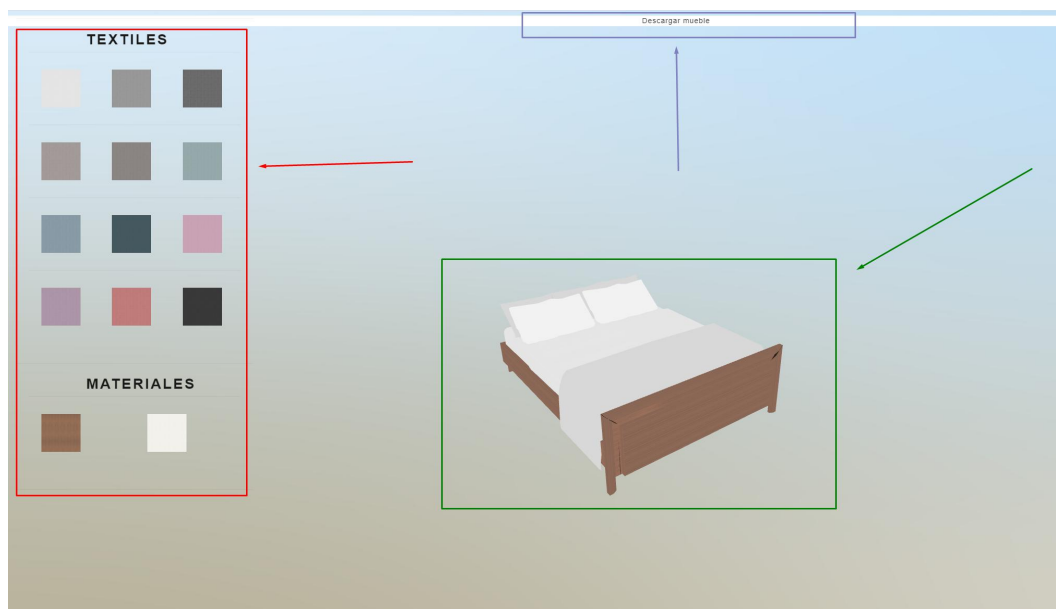


Figura B.2: Página del configurador 3D.

- Por último, se volvería a la página principal y en el formulario de encargo de muebles se rellenarían todos los campos y se adjuntaría la imagen descargada.

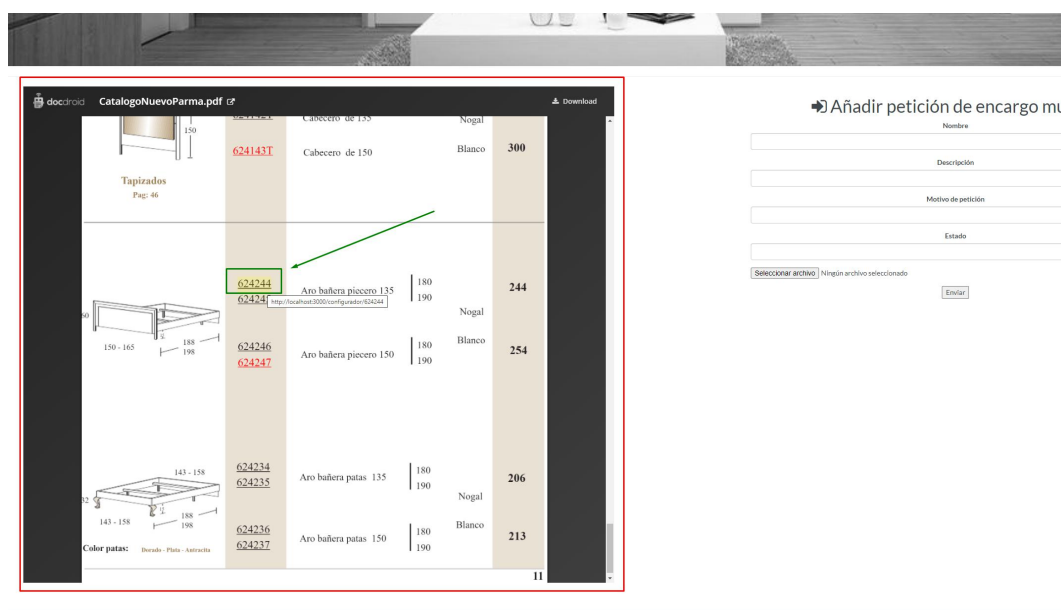


Figura B.3: Página principal de la aplicación (formulario de encargo).

- Si todo ha resultado se correcto saltará un mensaje indicando que el mueble se ha encargado con éxito.



## B.2. Manual administrador

- En la página principal se pincha al botón de acceso administrador.

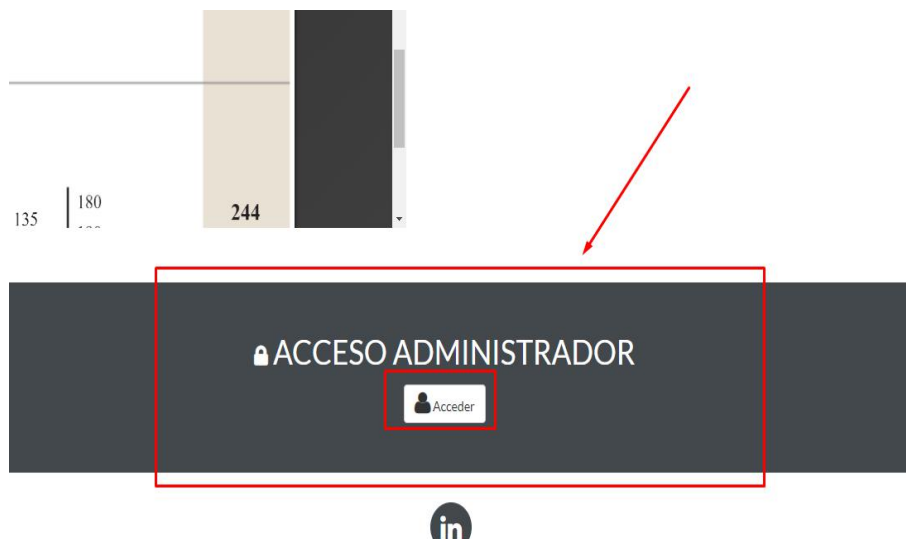


Figura B.4: Página principal de la aplicación (acceso administrador).

- En esta ventana se inicia sesión mediante usuario y contraseña.

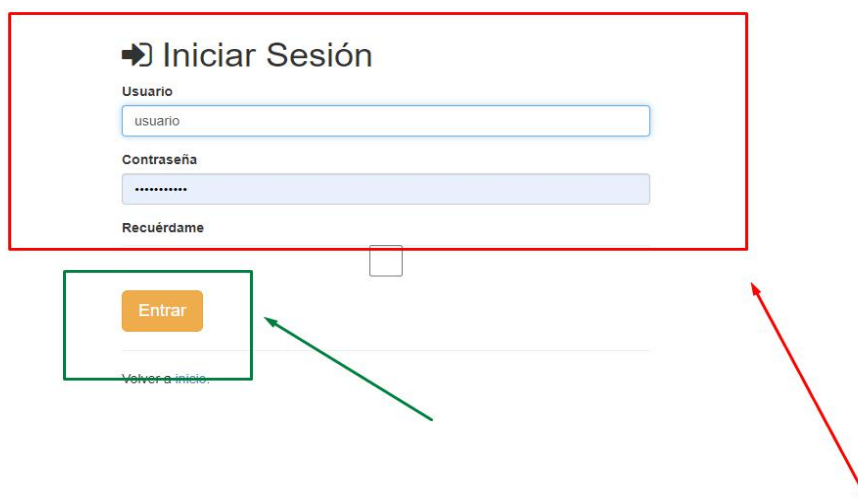


Figura B.5: Página de inicio de sesión.

- Si las credenciales eran correctas se accederá al área privada donde se puede observar información personal, los encargos realizados por clientes y se podrá crear un nuevo administrador en el botón inferior de la pantalla.

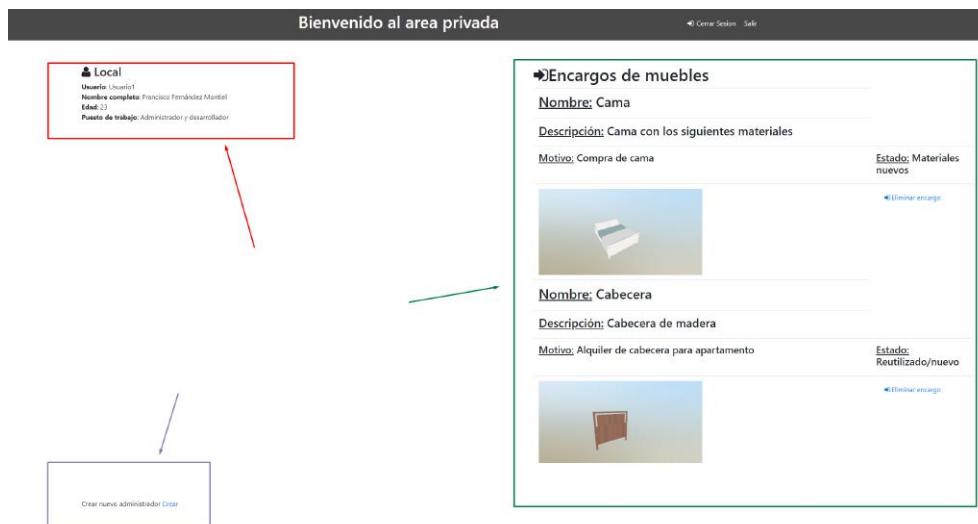


Figura B.6: Página del área privada de administradores.

- En esta ventana se rellenarán los datos de un nuevo administrador, indicando usuario y contraseña y pulsando el botón debajo del formulario se creará un nuevo perfil administrativo.

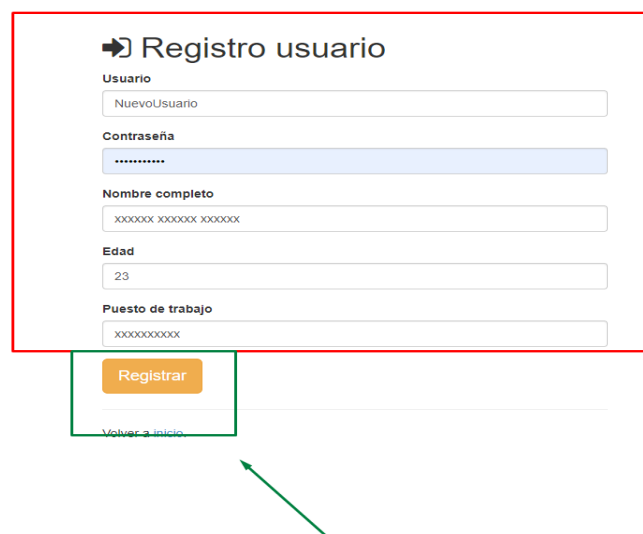


Figura B.7: Página de registro de nuevo administrador.

- Tras crear el nuevo administrador la página redirigirá nuevamente al área privada.



UNIVERSIDAD  
DE MÁLAGA

| **uma.es**

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática  
Bulevar Louis Pasteur, 35  
Campus de Teatinos  
29071 Málaga